
Masters Theses

Student Theses and Dissertations

1968

ZAP - a computer program to predict fission product concentrations as a function of neutron flux, irradiation and cooling times

Bruce Earl Koopmann

Follow this and additional works at: https://scholarsmine.mst.edu/masters_theses

 Part of the [Nuclear Engineering Commons](#)

Department:

Recommended Citation

Koopmann, Bruce Earl, "ZAP - a computer program to predict fission product concentrations as a function of neutron flux, irradiation and cooling times" (1968). *Masters Theses*. 5180.
https://scholarsmine.mst.edu/masters_theses/5180

This thesis is brought to you by Scholars' Mine, a service of the Missouri S&T Library and Learning Resources. This work is protected by U. S. Copyright Law. Unauthorized use including reproduction for redistribution requires the permission of the copyright holder. For more information, please contact scholarsmine@mst.edu.

ZAP - A COMPUTER PROGRAM TO PREDICT FISSION
PRODUCT CONCENTRATIONS AS A FUNCTION OF
NEUTRON FLUX, IRRADIATION AND COOLING TIMES

BY

BRUCE EARL KOOPMANN, /1968

A

THESIS

submitted to the faculty of

THE UNIVERSITY OF MISSOURI - ROLLA

in partial fulfillment of the requirements for the

Degree of

MASTER OF SCIENCE IN NUCLEAR ENGINEERING

Rolla, Missouri

1968

147512

Approved by

D. Day Edwards (advisor) Michael R. Kie
Herbert R. Alcorn

ABSTRACT

The development of a computer code was undertaken to calculate the fission product nuclide concentration for spent fuel discharged from a nuclear reactor, as a function of irradiation time, cooling time, neutron flux, and element cross section. The necessary input data was prepared from the basic nuclear data for fission of uranium-235 and plutonium-239. Calculations are also made for total or selective element beta heating and total gamma flux according to user selected input groups. These calculations would be useful in fuel element shielding, cooling, shipping and processing studies.

Acknowledgement

The author wishes to express his sincere appreciation to Dr. Doyle R. Edwards and Herbert R. Alcorn for their guidance and assistance in the preparation of this thesis.

TABLE OF CONTENTS

	Page
LIST OF FIGURES AND TABLES.....	v
I. INTRODUCTION.....	1
II. PRELIMINARY RESEARCH.....	2
III. THEORY.....	16
IV. APPLICATION.....	23
V. PROGRAM SUMMARY.....	25
VI. CONCLUSIONS.....	27
VII. RECOMMENDATIONS.....	29
REFERENCES CITED.....	31
APPENDIX 1. PROGRAM DESCRIPTION (ZAP)...	33
APPENDIX 2. PROGRAM LISTING.....	39
APPENDIX 3. DESCRIPTION OF INPUT.....	85
APPENDIX 4. EXAMPLE OF PROGRAM PROCEDURE.....	117
APPENDIX 5. FILE CONSTRUCTION.....	121
VITA.....	127

LIST OF FIGURES AND TABLES

FIGURES

	Page
1. Instantaneous Partial Fission Yield.....	10
2. Fission Yield of U235 Per 200 Fission Products.....	11

TABLES

I. Fission Yield of U235 (ca. 1 MeV Neutrons)/ 200 Atoms Fissioned.....	12
II. Fission Yield of PU239 (ca. 2 MeV Neutrons)/ 200 Atoms Fissioned.....	13
III. Fission Yield of U235 (Thermal Fissions)...	14
IV. Fission Yield of PU239 (Thermal Fissions)...	15

I. INTRODUCTION

Knowledge of the fission product spectrum, with respect to both atomic numbers and mass numbers, is often necessary in the reprocessing of nuclear fuel and the waste disposal industry. From this information, activity levels for shielding requirements and heat generation rates for cooling requirements may be calculated. These will depend upon irradiation time, cooling time, half-lives, cross sections, neutron energy, and decay energies as major factors.

At the present time, although calculation methods and equations are well known, the data required is far from complete. The data that is known, if compared from different reliable sources, may still vary. Although the nuclear decay equations and their solutions are relatively standard, the length and complexity of these equations lend themselves well to human error. The input data and number of equations to be solved represent a momentous task and the solution of such equations might very well prove economically unfeasible.

The technique described herein is incorporated in a computer program that solves such equations as mentioned above in a relatively short period of time with a minimum chance of human error.

II. PRELIMINARY RESEARCH

The importance of fission product concentration is ever present in almost all phases of the reactor industry. Glasstone and Sesonke (Ref. 1), Lamarsh (Ref. 2), Meghreblian and Holmes (Ref. 3) and others indicate this fact in their treatment of the different facets of the nuclear industry. Known fission product concentrations are necessary for deducing effects of reactor mishaps and fallout from nuclear weaponry. Reactor control is dependent on fission product buildup as is heat generation in spent fuel. Handling of spent and partially spent fuel, fuel reprocessing and waste disposal require knowledge of the activity of the subject material so that human dosage and processing techniques may be determined. Knowledge of fission product concentrations and fission product activity can not only help the nuclear industry be more efficient, but help save lives as well.

When reactors were in their infancy, fission product calculations were a long and tedious process requiring many man hours of computations involving the buildup and decay equations. These equations were generalized and put in symmetric form by Bateman (Ref. 4). These formulae were developed some time before the beginning of the nuclear age, and, although exact by theory, still required many hours of calculation and were subject to human error.

As early as 1950, graphical representations were given for general and specific irradiation and cooling times (Ref. 5). Graphical methods are still employed today, but are generally used for rough calculations rather than accurate answers. Later editions of these graphs also give average and total beta energy of fission products in addition to fission product concentrations (Ref. 6). As nuclear knowledge increased and digital computers came into being, hours of calculations were condensed to seconds with a minimum of human error. The more exact methods of calculating fission product concentrations became economically available to the user due to the great speed and relative availability of large computers. Some of the earlier programs written calculate only the fission product spectra. One, which might be called the predecessor of the program presented in this paper, was written by W.D. Owengs of Martin Company (Ref. 7), and calculates only the total fission product activity. Another program along this line is Delayne's method for specific calculations on individual isotopes (Ref. 8).

Later programs, such as FIP, written by G. Casadei (Ref. 9), numerically solve the differential equations governing the buildup and decay of fission products as a function of irradiation and decay times only. Another program along this line, by N. Harris of Atomics International (Ref. 10) obviously written for nuclear fallout

or reactor mishaps, calculates fission product buildup and decay and then calculates dose rates to various organs of the body due to inhalation of airborne contaminations.

Perhaps the best of the more recent programs in this area is a computer code developed by Oak Ridge Laboratory (Ref. 11) called PHOEBE. This code calculates fission product buildup and decay as a function of neutron flux, irradiation and cooling times. Functions are also developed for beta and gamma release rates. The derivation, usage, and modification for other purposes of these functions are explained in this report.

By no means are the references noted a complete list of available material. Literally hundreds of articles exist on different methods of calculating fission product buildup and decay, and beta and gamma release rates. Also, many computer programs exist using many of these methods and other shortened versions.

The Bateman equations are the basis of all decay and production of radioisotopes in decay chains. These equations, though simply solved, are tedious. They begin with the basic differential equation for decay to a daughter and proceed through a chain. Evans (Ref. 4) states that the solution of an accumulation of products is:

$$N = A_0 (h_a e^{-\lambda_a t} + h_b e^{-\lambda_b t} + \dots + h_m e^{-\lambda_m t} + h_n e^{-\lambda_n t}) \quad (1)$$

where:

N = number of atoms in decay daughter N after time t

$$h_a = \frac{\lambda_a}{\lambda_n - \lambda_a} \frac{\lambda_b}{\lambda_b - \lambda_a} \frac{\lambda_c}{\lambda_c - \lambda_a} \dots \frac{\lambda_m}{\lambda_m - \lambda_a}$$

$$h_b = \frac{\lambda_a}{\lambda_a - \lambda_b} \frac{\lambda_b}{\lambda_n - \lambda_b} \frac{\lambda_c}{\lambda_c - \lambda_b} \dots \frac{\lambda_m}{\lambda_m - \lambda_b}$$

$$h_m = \frac{\lambda_a}{\lambda_a - \lambda_m} \frac{\lambda_b}{\lambda_b - \lambda_m} \frac{\lambda_c}{\lambda_c - \lambda_m} \dots \frac{\lambda_m}{\lambda_n - \lambda_m}$$

$$h_n = \frac{\lambda_a}{\lambda_a - \lambda_n} \frac{\lambda_b}{\lambda_b - \lambda_n} \frac{\lambda_c}{\lambda_c - \lambda_n} \dots \frac{\lambda_m}{\lambda_m - \lambda_n}$$

Equation (1) can be stated in a more general sense as follows: The number of atoms present of a given nuclide (N) after any time (t) is equal to the number of atoms present at the beginning of the chain (A_0) times a sum of probabilities of decay. The probabilities of decay is the probability that the atom will decay ($e^{-\lambda_j t}$) times the probability that an atom belongs to this decay daughter (h), i.e., the fraction of A_0 decaying by this mode. This could be made more general by letting, "probability that

something will happen to the atom", replace, "probability that the atom will decay".

A_0 need not be the number of atoms beginning the chain, but may be broken down into the number of atoms present at each step of the chain (Y).

If we now follow closely the general solution of the original decay equation letting each element be removed (μ) instead of just decaying (λ) but considering the parent only, decays to the daughter, then $e^{-\lambda_j t}$ becomes $e^{-\mu_j t}$ and h_j remains with μ 's replacing λ 's in denominator.

The Bateman solutions may then be written as follows

$$N = Y_a h_a P_a + Y_b h_b P_b + \dots + Y_m h_m P_m + Y_n h_n P_n \quad (2)$$

where: Y_j = yield of a node in a decay chain and is a function of all Y 's and h 's whose subscript is less than j .

h_j = has the same form except lambda in the denominator becomes mu's with same subscripts and since Y_j is a function of $h_{x>j}$, h_j only contains parameters from j to n .

P_j = probability of some type of change (i.e., $e^{-\mu_j t}$ for decay, $1 - e^{-\mu_j t}$ for no decay).

Remembering that the rate of production for an element is Y_i/μ_i from the previous element, using the step ahead technique we may generalize the Bateman equations as is

done in the next section.

The complex interrelationships between nucleons when they form medium and heavy nuclei will probably continue to defy precise analysis for a long time to come. In the absence of exact theory, a number of nuclear models have been developed, each utilizing a different set of simplifying assumptions. Each model is capable of explaining a portion of experimental knowledge about nuclei. The two models considered were the Nuclear Shell Model and the Liquid Drop Model.

The Nuclear Shell Model, in contrast with the situation with atoms, assumes the nucleus contains no massive central body which can act as a force center. It also assumes that each nuclide experiences a central attractive force due to the other nucleons present. This allows one to think of each nucleon as moving independently in a cloud with little interaction of other nucleons. From this model (as described in Ref. 4) we obtain the theoretical calculations of the most stable atomic number for a specified mass number.

The Liquid Drop Model, originated in Bohr's concept of the compound nucleus in nuclear reactions, gives semi-empirical forms predicting decay energies, binding energies, and energetics of fission. This model assumes that interactions between nucleons is strong. Levels of energy

are quantized states of the system rather than states of individual particles. Energies of captured particles appear to be shared by target and captured particles almost instantaneously and these particles have a small mean free path. From this model (as described in Ref. 4) we obtain the semi-empirical formulae for beta decay.

It is interesting to note that Geilikman (Ref. 12) states that the Liquid Drop Model and Nuclear Shell Model are not incompatible for fission analysis and gives cases in point.

For the so called "removal" constant, μ , substituted in the Bateman equations we have $\mu = \lambda + \phi\sigma$. Cross sections (σ) depend not only on the element considered, but also on neutron properties. Westcott effective cross sections (Ref. 13) were chosen to be included in this paper due to their simplicity and good accuracy. Westcott defined his cross section in terms of neutron spectrum. If this spectrum is present, the effective cross section at a desired temperature is simply a multiple of the 2200 meter cross sections. Another method considered was the statistical method for fission products by the GUNYA programs as outlined in Ref. 14. This method is reasonably accurate, according to the author, but since it entails programming and programs in itself, it was not chosen. Other available methods were considered but for ease of calculation

both by the author and user, Westcott effective cross sections were thought most appropriate.

To perform the necessary calculations, it is necessary to know the instantaneous fission yield spectrum as a function of mass number (A) and atomic number (Z); that is, the instantaneous partial fission yield $Y(Z,A)$ is desired. This three dimensional surface is sketched in Figure 1, and experimental values may be taken directly from Ref. 15, or calculated as follows.

For a given mass number, the distribution of fission products with respect to atomic number is approximately a Gaussian distribution centered about the most probable atomic number, $Z_p(A)$. These curves are shown as slices parallel to the Y,Z plane in Figure 1. A typical instantaneous fission yield curve, or spectrum, as a function of mass number (A) is shown in Figure 2 and tabulated in Tables I through IV taken from Ref. 15 for the four different types of fissions considered in this paper. Figure 2 corresponds to a view of Figure 1 parallel to the Z axis. From the total mass yield curve $Y(A)$ and these Gaussian distributions, the instantaneous partial fission yield surface $Y(Z,A)$ may be constructed. This surface will give the probability of producing a nuclide (${}_Z^AX^A$) from a fission.

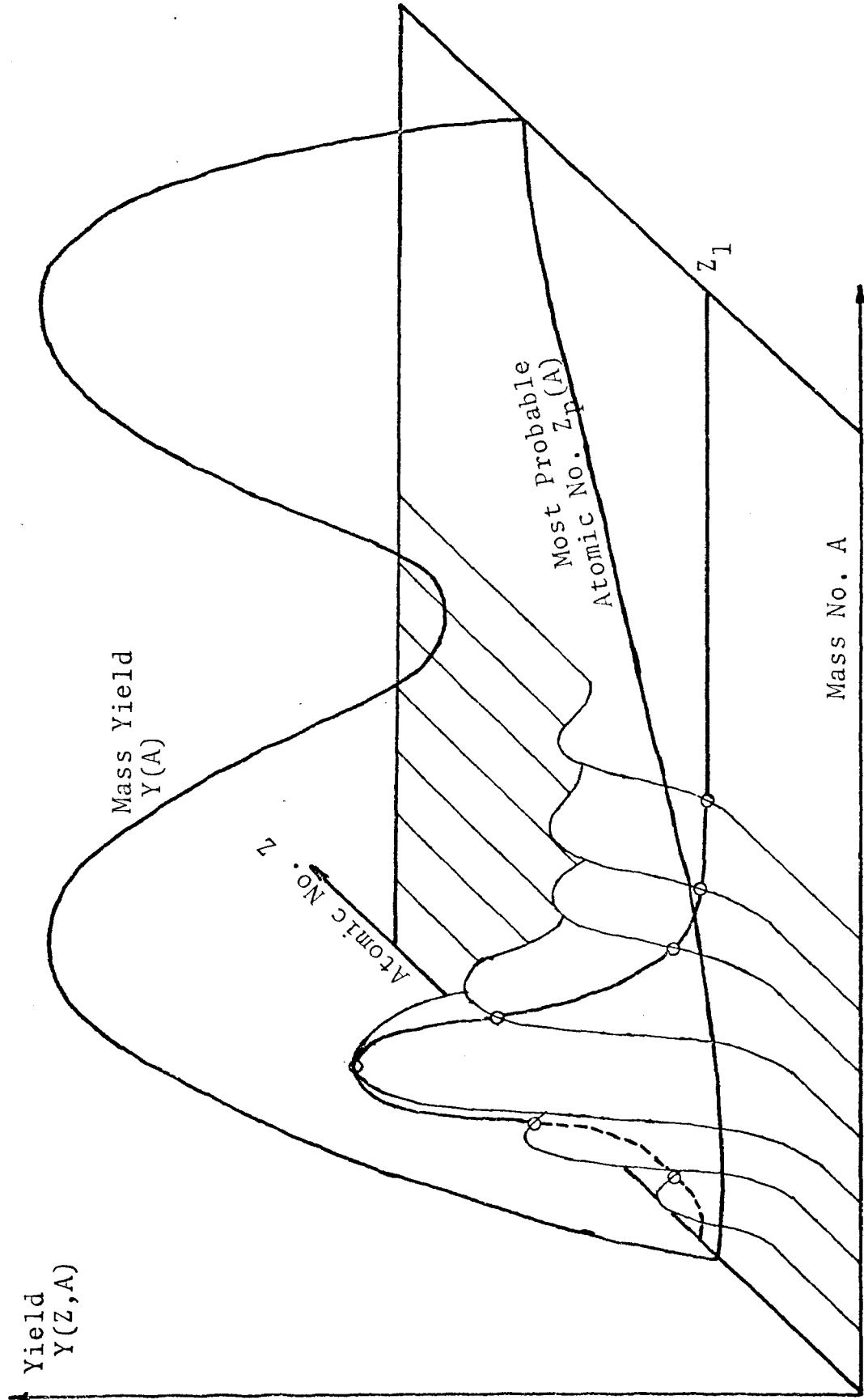


Fig. 1. Instantaneous Partial Fission Yield.

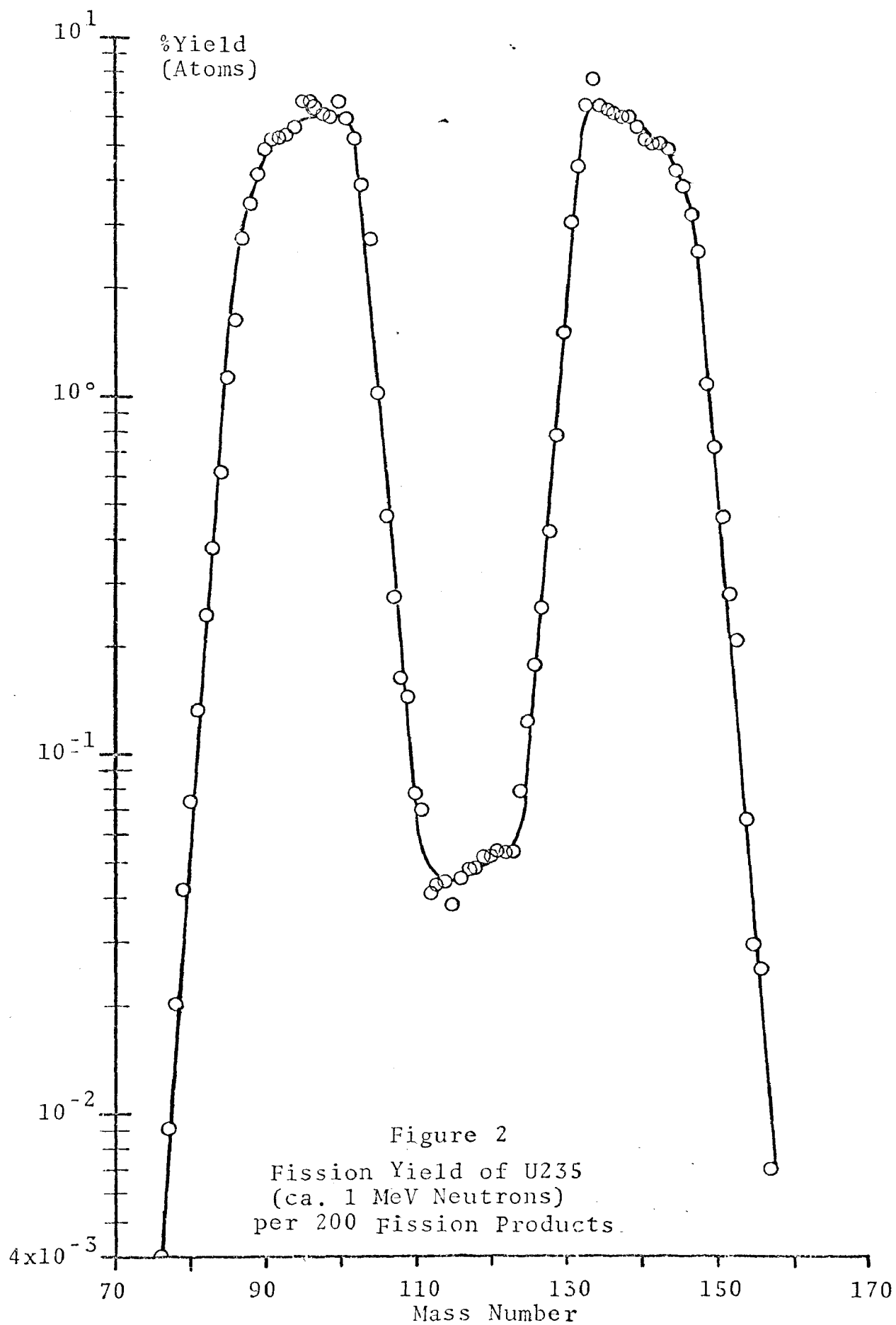


Table I
Fission Yield of U235
(ca. 1 MeV Neutrons)/200 Atoms Fissioned

Mass	Yield	Mass	Yield	Mass	Yield
70	.000E 00	100	.668E 01	130	.151E 00
71	.000E 00	101	.599E 01	131	.308E 01
72	.000E 00	102	.525E 01	132	.440E 01
73	.000E 00	103	.396E 01	133	.655E 01
74	.990E-03	104	.274E 01	134	.773E 01
75	.198E-02	105	.101E 01	135	.649E 01
76	.396E-02	106	.465E 00	136	.636E 01
77	.891E-02	107	.277E 00	137	.624E 01
78	.198E-01	108	.161E 00	138	.606E 01
79	.416E-01	109	.144E 00	139	.619E 01
80	.743E-01	110	.782E-01	140	.574E 01
81	.131E 00	111	.703E-01	141	.525E 01
82	.247E 00	112	.406E-01	142	.514E 01
83	.378E 00	113	.425E-01	143	.514E 01
84	.619E 00	114	.435E-01	144	.495E 01
85	.113E 01	115	.376E-01	145	.427E 01
86	.162E 01	116	.445E-01	146	.383E 01
87	.286E 01	117	.475E-01	147	.320E 01
88	.346E 01	118	.475E-01	148	.225E 01
89	.414E 01	119	.515E-01	149	.108E 01
90	.495E 01	120	.515E-01	150	.728E 00
91	.526E 01	121	.544E-01	151	.460E 00
92	.525E 01	122	.525E-01	152	.279E 00
93	.537E 01	123	.525E-01	153	.208E 00
94	.564E 01	124	.792E-01	154	.663E-01
95	.670E 01	125	.121E 00	155	.287E-01
96	.668E 01	126	.178E 00	156	.247E-01
97	.649E 01	127	.257E 00	157	.693E-02
98	.618E 01	128	.425E 00	158	.000E 00
99	.604E 01	129	.782E 00	159	.000E 00
				160	.000E 00

Yield Sum = 199.9999

Table II
Fission Yield of PU239
(ca. 2 MeV Neutrons)/200 Atoms Fissioned

Mass	Yield	Mass	Yield	Mass	Yield
70	.000E 00	100	.585E 01	130	.292E 01
71	.000E 00	101	.604E 01	131	.477E 01
72	.000E 00	102	.619E 01	132	.622E 01
73	.000E 00	103	.627E 01	133	.609E 01
74	.000E 00	104	.643E 01	134	.614E 01
75	.000E 00	105	.470E 01	135	.606E 01
76	.000E 00	106	.619E 01	136	.585E 01
77	.000E 00	107	.468E 01	137	.663E 01
78	.000E 00	108	.321E 01	138	.546E 01
79	.000E 00	109	.165E 01	139	.516E 01
80	.000E 00	110	.819E 00	140	.487E 01
81	.000E 00	111	.477E 00	141	.458E 01
82	.780E-02	112	.180E 00	142	.429E 01
83	.955E-01	113	.165E 00	143	.477E 01
84	.195E 00	114	.123E 00	144	.360E 01
85	.321E 00	115	.916E-01	145	.316E 01
86	.516E 00	116	.780E-01	146	.282E 01
87	.789E 00	117	.682E-01	147	.248E 01
88	.123E 01	118	.682E-01	148	.214E 01
89	.179E 01	119	.682E-01	149	.185E 01
90	.224E 01	120	.682E-01	150	.154E 01
91	.270E 01	121	.877E-01	151	.126E 01
92	.321E 01	122	.107E 00	152	.877E 00
93	.365E 01	123	.124E 00	153	.468E 00
94	.409E 01	124	.175E 00	154	.292E 00
95	.513E 01	125	.253E 00	155	.175E 00
96	.497E 01	126	.390E 00	156	.780E-01
97	.507E 01	127	.604E 00	157	.195E-01
98	.546E 01	128	.107E 01	158	.000E 00
99	.575E 01	129	.175E 01	159	.000E 00
				160	.000E 00

Yield Sum = 199.9999

Table III
Fission Yield of U235
(Thermal Fissions)

Mass	Yield	Mass	Yield	Mass	Yield
70	.000E 00	100	.632E 01	130	.200E 00
71	.000E 00	101	.501E 01	131	.288E 01
72	.160E-04	102	.411E 01	132	.432E 01
73	.110E-03	103	.300E 01	133	.651E 01
74	.351E-03	104	.180E 01	134	.792E 01
75	.381E-03	105	.902E 00	135	.633E 01
76	.100E-02	106	.381E 00	136	.638E 01
77	.802E-02	107	.190E 00	137	.606E 01
78	.210E-01	108	.822E-01	138	.575E 01
79	.561E-01	109	.300E-01	139	.657E 01
80	.802E-01	110	.220E-01	140	.646E 01
81	.140E 00	111	.190E-01	141	.601E 01
82	.290E 00	112	.100E-01	142	.596E 01
83	.545E 00	113	.100E-01	143	.599E 01
84	.100E 01	114	.110E-01	144	.568E 01
85	.130E 01	115	.110E-01	145	.396E 01
86	.202E 01	116	.110E-01	146	.308E 01
87	.249E 01	117	.110E-01	147	.238E 01
88	.358E 01	118	.120E-01	148	.170E 01
89	.480E 01	119	.130E-01	149	.113E 01
90	.578E 01	120	.140E-01	150	.672E 00
91	.585E 01	121	.150E-01	151	.451E 00
92	.604E 01	122	.160E-01	152	.285E 00
93	.647E 01	123	.170E-01	153	.150E 00
94	.642E 01	124	.190E-01	154	.772E-01
95	.629E 01	125	.210E-01	155	.331E-01
96	.635E 01	126	.501E-01	156	.140E-01
97	.611E 01	127	.130E 00	157	.802E-02
98	.579E 01	128	.371E 00	158	.200E-02
99	.607E 01	129	.902E 00	159	.100E-02
				160	.461E-03

Yield Sum = 200.0000

Table IV
Fission Yield of PU239
(Thermal Fissions)

Mass	Yield	Mass	Yield	Mass	Yield
70	.000E 00	100	.708E 01	130	.249E 01
71	.000E 00	101	.589E 01	131	.376E 01
72	.129E-03	102	.597E 01	132	.508E 01
73	.249E-03	103	.565E 01	133	.678E 01
74	.349E-03	104	.591E 01	134	.738E 01
75	.149E-02	105	.549E 01	135	.698E 01
76	.199E-02	106	.455E 01	136	.668E 01
77	.378E-02	107	.339E 01	137	.661E 01
78	.159E-01	108	.197E 01	138	.629E 01
79	.219E-01	109	.139E 01	139	.585E 01
80	.688E-01	110	.599E 00	140	.548E 01
81	.124E 00	111	.229E 01	141	.528E 01
82	.196E 00	112	.119E 00	142	.449E 01
83	.373E 00	113	.598E 01	143	.468E 01
84	.468E 00	114	.508E 01	144	.388E 01
85	.664E 00	115	.408E-01	145	.312E 01
86	.757E 00	116	.378E-01	146	.259E 01
87	.917E 00	117	.369E-01	147	.194E 01
88	.141E 01	118	.359E-01	148	.172E 01
89	.170E 01	119	.359E-01	149	.137E 01
90	.224E 01	120	.359E-01	150	.100E 01
91	.260E 01	121	.428E-01	151	.797E 01
92	.313E 01	122	.458E-01	152	.618E 01
93	.395E 01	123	.498E-01	153	.369E 01
94	.446E 01	124	.797E-01	154	.289E 00
95	.501E 01	125	.119E 00	155	.194E 00
96	.515E 01	126	.194E 00	156	.109E 00
97	.563E 01	127	.388E 00	157	.807E-01
98	.587E 01	128	.997E 00	158	.418E-01
99	.608E 01	129	.197E 01	159	.209E-01
				160	.897E-01

Yield Sum = 200.0000

III. THEORY

The buildup and decay of a radioactive nuclide (i.e. burnup time) is governed by the first order differential equation:

$$\frac{dC_i}{dt} = \lambda_{i-1} C_{i-1} + R_i - \mu_i C_i \quad (3)$$

where:

C_i = concentration of the i'th nuclide in a straight line decay chain

λ_i = decay constant

R_i = production rate

$i-1$ = parent of i'th nuclide

$\mu_i = \lambda_i + \phi \sigma_i$ = removal rate constant

where:

ϕ = neutron flux

σ_i = microscopic cross section for neutron absorption

Obviously for the period of time after reactor shutdown (i.e. cooling time), $R_i=0$ and $\mu_i=\lambda_i$. Equation (3) reduces to

$$\frac{dC_i}{dt} = \lambda_{i-1} C_{i-1} - \lambda_i C_i \quad (4)$$

Equations (3) and (4) may be solved by conventional techniques with a step ahead technique from the first nuclide in a straight line decay chain giving a condensed form of the familiar Bateman equations. Using the initial

condition of zero concentration, the concentration of the i'th nuclide (N_i) in the decay chain after burnup time (T) from equation (3) is:

$$N_i(T) = \sum_{m=1}^i Y_m \left\{ \frac{Q}{T} r_m \left[\prod_{\ell=m}^{i-1} \lambda_{\ell} \right] \left[\sum_{j=m}^i \frac{1 - e^{-\mu_j T}}{\mu_j \prod_{\substack{K=m \\ K \neq j}} (\mu_K - \mu_j)} \right] \right\} \quad (5)$$

where:

Y_m = yield fraction of m'th nuclide

Q = basis of calculation, number of nuclei fissioned

r_m = equivalent straight line decay branching fraction

For the last nuclide in the decay chain $\lambda_{\max} = 0$ and application of L'Hospitals rule yields the following for the terms in braces in equation (5).

$$\left\{ \frac{Q}{T} r_m \left[\prod_{K=m}^{i-1} \lambda_K \right] \left[\sum_{j=m}^{i=1} \frac{1 - e^{-\mu_j T}}{\mu_j \prod_{\substack{K=m \\ K \neq j}} (\mu_K - \mu_j)} \right] + r_m Q \right\}$$

when: $i = \text{IMAX}$ $i \neq m$

or $\{r_m Q\}$

when $i=m=\text{IMAX}$

where IMAX denotes last member of a chain.

Using equation (5) for initial conditions, equation (4) may be solved in similar manner to obtain the concentration of the i 'th nuclide (n_i) in the decay chain after cooling time (t):

$$n_i(t) = \sum_{m=1}^i N_m(T) \left[\prod_{\ell=m}^{i-1} \lambda_{\ell} \right] \left[\sum_{j=m}^i \frac{e^{-\lambda_j t}}{\prod_{\substack{K=m \\ K \neq j}}^i (\lambda_K - \lambda_j)} \right] \quad (6)$$

The instantaneous fission yield percentages (Y_m) used were taken from Refs. 15, 16 and 17, and tabulated in Tables I through IV. Yields are given by mass number (A) only and form the familiar "M" shaped curve of fission products shown in Figure 2. No information is given here with respect to atomic number (Z).

The most stable atomic number for a specified mass number is taken from Ref. 18 and is a result of theoretical calculations based upon the Nuclear Shell Model. Use is then made of the fact that the deviation of the most probable atomic number from the most stable atomic number is the same for both the light and heavy fission fragments to calculate the most probable atomic number for both the heavy and light fission fragments. The terms heavy and light fission fragments are arbitrary; they refer to the upper and lower halves respectively of the fission product yield curve for mass numbers. Calculations were

performed using the relation:

$$Z_{p\ell} = \frac{1}{2} [Z_f - (Z_{sh} - Z_{s\ell})] , \quad Z_{ph} = Z_f - Z_{p\ell} \quad (7)$$

where:

Z_f = atomic no. of fissile material

Z_{sh} = most stable atomic number for high mass
number (A)

$Z_{s\ell}$ = most stable atomic number for low mass
number (A)

Z_{ph} = most probable atomic number for high mass
number (A)

$Z_{p\ell}$ = most probable atomic number for low mass
number (A)

The atomic number has been found (Ref. 18 and 19) from experimental data to have an empirical Gaussian distribution about the most probable atomic number, for a specified mass number, of the form:

$$P(Z) = \frac{1}{\sqrt{\pi}} e^{-(Z-Z_p)^2} \quad (8)$$

From the relations (7) and (8), and yield data, the instantaneous partial fission product yields may be calculated by:

$$Y(Z,A) = Y(A)P(Z) = Y(A) \frac{1}{\sqrt{\pi}} e^{-[Z-Z_p(A)]^2} \quad (9)$$

where:

$Y(A)$ = absolute yield per mass number

$Y(Z,A)$ = absolute yield per mass number and atomic number

$P(Z)$ = probability of distribution about Z_p

Z = atomic number in question for specified mass number

Z_p = most probable atomic number for specified mass number

Since the partial yields for a given mass number (A) must sum to the total yield, any partial yield is not accounted for, and is assigned to the progenitor of that mass number. Thus:

$$Y_{\text{prog}}(A) = Y(A) - \sum_i Y(Z_i, A) \quad (10)$$

The progenitor corresponds to all nuclides that are so unstable they decay almost instantaneously by β^- emission to the first member of the decay chain with a measurable half life. They are therefore added to the first member of their respective decay chains.

For nuclides with two different isomeric states, metastable and normal, or three different isomeric states, A, B, and metastable, the partial yield is divided equally between the states.

Evans (Ref. 4) states that semi-empirical equations

exist which predict the average beta energy released in radioactive decay based on the liquid drop model. These are:

$$Q_{\beta} = 2\gamma [\pm(Z_0 - Z) - \frac{1}{2}] \quad \text{for odd } A \quad (11)$$

$$Q_{\beta} = 2\gamma [\pm(Z_0 - Z) - \frac{1}{2}] \left\{ \begin{array}{l} + 2\delta \text{ for odd } Z \\ - 2\delta \text{ for even } Z \end{array} \right\} \text{for even } A \quad (12)$$

where:

\pm = chosen according $Z \rightarrow (Z \pm 1)$

Q_{β} = energy of released electron (MeV)

Z = atomic number in question

Z_0 = most stable isobar for specified mass number

δ = pairing energy (MeV)

$$\gamma = \frac{4a_a}{A} \left(1 + \frac{A^{2/3}}{4a_a/a_c} \right)$$

$$a_c = (.595 \pm .02) \text{ MeV}$$

$$\frac{a_a}{a_c} = \frac{1}{2} \left\{ \frac{Z_0 A^{2/3}}{A - 2Z_0} \right\} - \left\{ \frac{A}{A - 2Z_0} \right\} \frac{(M_n - M_H)}{4a_c} \approx 32 \pm 1 (\text{average})$$

It will be noted that pairing energy is not included in the liquid drop model development. As is suggested by Evans, a smooth function of δ was chosen, i.e. $\delta \approx \frac{C}{A^x}$.

For the author's development, average values were chosen for the entire spectrum. They are:

$$\gamma_{av} = \frac{76}{A} \left(1 + \frac{A^{2/3}}{128} \right) \text{ MeV}$$

$$Z_o = \frac{77}{2\gamma_{av}}$$

$$\delta_{av} = \frac{33.5}{A^{3/4}} \text{ MeV}$$

Semi-empirical or empirical equations are available to predict average gamma release from radioactive decay. The author feels that due to the very complex nature of these equations, exact determinations of gamma decay energies would be best left to a separate project. For the purposes of data for this paper, gamma decay energies were used as follows:

Given Q_β as determined by equation (11) or (12)

$$\text{If } |Z - Z_o| \leq 1: \quad \text{then} \quad Q_\gamma = .15Q_\beta \text{ (MeV)} \quad (13)$$

$$\text{If } 1 \leq |Z - Z_o| \leq 2: \quad \text{then} \quad Q_\gamma = .55Q_\beta \text{ (MeV)} \quad (14)$$

$$\text{otherwise} \quad Q_\gamma = .5 \text{ (MeV)} \quad (15)$$

IV. APPLICATION

The actual intended purpose of this work was to develop a "single package type" computer program whereby a minimum of input would be required by a user and a maximum number of options may be chosen. The author believes this has been accomplished with a type of program control that is, whenever possible, completely alphabetic.

The program, described in Appendix 1 and listed in Appendix 2, once generated, requires a minimum of program control input, as described in Appendix 3.

Available mass yield data is as varied as there are types of operating reactors. Among other variables, type of fuel and energy of neutrons causing fission determine fission product spectra. Four curves were chosen as standards, they are: fissioning of the U-235 by thermal and 1 MeV neutrons, and Pu-239 by thermal and 2 MeV neutrons. However, any of these curves may be altered or deleted and others inserted in their place.

Neutron cross sections are only somewhat varied, but are not complete for all isotopes. The cross sections used by this paper were taken from BNL-325 (Ref. 20) whenever possible and from Chart of the Nuclides (Ref. 21) otherwise.

As stated above, the most stable atomic number for a

specified mass number was calculated by semi-empirical formulas taken from Refs. 18 and 19.

Nuclear decay chains are many and varied. The fission product decay chains have been summarized for uranium and plutonium in Ref. 22 and have been used in preparing this paper. A sample problem explaining breakdown of complex chains to simple straight line decay chains and ensuing calculations is included in Appendix 4.

A breakdown of the files used is included in Appendix 5. Input and output of a sample run are included in Appendix 3.

V. PROGRAM SUMMARY

The entire assembled program is named ZAP and is divided into 4 major steps. ZAP will, beginning with basic data, supply and make resident all work files, read and interpret input data, do calculations and output specified options, and rewrite all data sets to tape, or holding area, and free resident work files. More specifically:

1. Step 1 - STARTZAP

This step reads files prepared by the author (or subsequent files after alteration by user) and creates working files on units specified by the user. Caution must be taken as two data sets are direct access or update type of files, consequently, at least these two data sets must reside on direct access volumes.

2. Step 2 - ZAP

This step reads and interprets program input data and passes this data on to step 3. It also checks positioning of characters, notes errors detected, informs user of the number and location of errors, and suggests to the user whether or not to proceed.

3. Step 3 - SUPERZAP

This step does the actual input, calculations, and output according to information from step 2.

This step is broken into ten major sections, and a detailed description of each section may be found in Appendix 1.

4. Step 4 - ENDZAP

This step rewrites work files to tape or other device specified by the user and deletes work files.

Between any of these steps, permanent files may be altered, deleted, or completely rewritten using the support utilities described in Appendix 1 and listed in Appendix 2; or by the user writing his own routines following the file construction format listed in Appendix 5.

VI. CONCLUSIONS

The intention of this paper was to create and make usable a computer program that would incorporate as much useful output as possible concerning fission product build-up and decay with a minimum of input and man hours of work; this, the author believes, has been accomplished.

A number of program options have been included covering type of input and type of output desired. The user may specify cooling and burnup times (except infinite) as desired. He may choose yields specified completely by the program, input his own mass yield curve and use the program specified distribution, or input his own yields per mass number and atomic number. He may choose to: negate neutron burnup of fission products by using $\mu=\lambda$ instead of $\mu=\phi\sigma+\lambda$ in equation (5), leave μ unchanged, let $\mu=\lambda+\phi\sigma$ and choosing between Westcott effective cross sections or input his own cross-sections. Many more options are available and are listed and explained in Appendix 3.

It is possible to minimize input to 9 cards or make the input as extensive as necessary to cover a particular problem. It is entirely possible that a nuclear power company or similar facility, after alteration of some permanent files, may calculate how much of any isotope is present in their reactor after any time specified. The program

will also predict individual and total beta energy, and photon energies by group up to 80 groups.

All this information can be obtained, if chosen, by semi-empirical means rather than individual equations and human mechanics. The savings in time, human effort, and cost will be noteworthy.

VII. RECOMMENDATIONS

Although a number of options have been included in this initial coding, others may be introduced. Some suggestions are:

1. As the code now stands isotope concentrations are calculated on the basis that the reactor is being started for the first time, i.e. the concentration of fission products at program run time is zero. An option may be included to input concentrations, or take them from the output of the last run, and include them in the present run to predict concentrations for any number of burnup and cooling times for the same reactor.
2. An option may be included for multiple outputs such that more than one copy may be obtained at once. At the present time, only one copy of output is attainable when the program is run. To obtain multiple copies, subroutines OUTPUT, SPCASE HEAT, and/or SHIELD have to be run independently.
3. More options, for prediction of partial yields from other sources than listed in Appendix 3.
4. Different methods of calculating effective cross sections.
5. Different methods of calculating and/or inputting average beta and gamma decay energies.

6. A general update program may be written to update any or all permanent or work files.

The author feels that a better method for predicting average beta and gamma decay energies could be devised and would recommend that this area be investigated first.

REFERENCES

1. Glasstone, S., and A. Sesonske (1963) Nuclear Reactor Engineering, D. Van Nostrand Co, Inc, New York, 830 p.
2. Lamarsh, J.R. (1966) Nuclear Reactor Theory, Addison-Wesley, Massachusetts, 585 p.
3. Meghreblian, R.V., and D.K. Holmes (1960) Reactor Analysis, McGraw-Hill, New York, 808 p.
4. Evans, R.D. (1955) The Atomic Nucleus, McGraw-Hill, New York, 972 p.
5. Cook, G.B., and F. Morgan (1950), The Growth, Decay and Beta Absorption Characteristics of the Long-Lived Fission Products, (Atomic Energy Research Establishment, Harwell, England), AERE-C/M-58, 10 p.
6. MacBean, I.J. (1960) Uranium-235 Fission Product Decay Energies, (Atomic Energy Research Establishment, Winfrith, England), as quoted in Nuclear Power 5, No. 49, p. 115-117.
7. Owings, W.D. (1959) A Program for the Computation of Fission Product Activity, MND-1721, 45 p.
8. Delaney, J.A. (1959) Digital Analysis of Fission Product Buildups and Decay, DC59-11-85, 61 p.
9. Casadei, G., G. Cenacchi, A. DeMatteis, and B. Faleschine (1964) FIP- An IBM-7094 Program for Calculation of Concentrations of Fission Products, RT/FIMA(64)-3, Rome, Italy, 26 p.
10. Harris, A., P. Kenfield, W. Laks, and W. Sayer (196?), An Improved Method of Dose Calculation from Airborne Fission Products, CONF-642-8, 28 p.
11. Arnold, E.D. (1966) PHOEBE: A Code for Calculating Beta and Gamma Activity and Spectra for Uranium-235 Fission Products, ORNL-3931, 64 p.

12. Geilikman, B.T. (1964) Shell Effects in Nuclear Fission
As translated by Elmar K. Wilip (Argonne National
Laboratory) from IAE-818, ANL-Trans-243, 4 p.
13. Westcott, C.H. (1958) Effective Cross Section Values for
Well Moderated Thermal Reactor Spectra, AECL-670, 27 p.
14. Cook, J.L (1966) Statistical Calculations of Fission Pro-
duct Cross-Sections as quoted in Nuclear Data for
Reactors - Volume One, IAEA, Vienna 1967, 9 p.
15. Koopman, B.E., and T.R. Hill (1967) Extension of a Com-
puter Program for Fission Product Spectra in Fuel
Irradiated with One and Two MeV Neutrons (II), AMU-
ANL Report, Argonne National Laboratory, 34 p.
16. Koopman, B.E., G.D. Atkinson, and T.R. Hill (1967) Exten-
sion of a Computer Program for Fission Product Spectra
in Fuel Irradiated with One and Two MeV Neutrons (I),
AMU-ANL Report, Argonne National Laboratory, 105 p.
17. Burris, L., and I. Dillon (July 1957) Estimation of Fission
Product Spectra in Discharged Fuel from Fast Reactors,
ANL 5732 (Revised 1963 by ANL-LB-SL-2410).
18. Hyde, E.K. (Jan 1960) A Review of Nuclear Fission - Part
One, UCRL-9030.
19. Steinberg, E.P., and L.E. Glendenin (1955) Proceedings of
the International Conference on the Peaceful Uses of
Atomic Energy - Volume 7, United Nations, Geneva,
p. 614.
20. Hughes, D.J. and R.B. Swartz (1958) Neutron Cross Sections,
2nd Edition, Brookhaven National Laboratory (BNL-325),
New York. As ammended by BNL-325 Suppliments (1966),
Brookhaven National Laboratory.
21. Goldman, D.T. and J.R. Rosser (1960) Chart of the Nuclides,
General Electric Co., New York.
22. Crocker, G.R. (1967) Fission Product Decay Chains: Sche-
matics with Branching Fractions, Half-Lives, and
Literature References, AD-659-980, 30 p.

APPENDIX 1

Program Description (ZAP)

STARTZAP

This routine reads such files from magnetic tape or other storage unit as is specified (See Appendix 3 for input format), and makes them resident on work files. The routine is extremely simple consisting of essentially all reads and writes and is very easy to follow (See Appendix 2 for listing).

ZAP

This routine essentially reads, interprets, and writes input to SUPERZAP. The entire text of any one card is not checked. Consequently, misspelled words will not necessarily cause errors. In general, column one is checked to determine the validity of the card. If confusion could result from two adjacent cards having the same character in column one, other characters are checked to determine correctness.

Once a valid card is determined, equal signs or commas are checked to ensure that numerical data is in proper columns. Each time a new card is read, column one is checked for a dollar sign denoting a comment.

Finally, data is tabulated and written to disk to later be picked up by SUPERZAP from files 9 and 11. Cards read are printed under column headings to help facilitate

error analysis. If cards are properly written, notation is printed specifying options chosen.

This routine calls ERR and REREAD as listed in this appendix under utilities.

SUPERZAP

This routine does the actual calculations and final output, and is broken into ten major subroutines. They are, with their purposes:

MAIN. This section reads data written by ZAP, places proper variables in common storage, and serves to call other subprograms as many times as there are burnup times.

TIME. This subroutine converts input cooling and burnup times to seconds for use in the other subprograms.

DTAPRP. This subroutine prepares work file 4 from files 14 through 18 or inputs files (See Appendix 5 for file construction) with yield data or input data as specified by ZAP. This routine prints the progenitor and oblong tables as specified. (See Theory, equations (7), (8), (9), and (10)).

CROSS. This subroutine calculates μ as specified by ZAP. This subprogram also prints μ , λ , and/or cross sections with neutron flux as specified by ZAP. This information is written to file 4 completing its construction (See Theory, equation (3)).

ZEROF5. This subprogram sets previous cooling time answers to zero throughout file 5, (See recommendations for suggested changes).

CALC. This subprogram reads file 4, does the actual calculation concerning final concentrations and updates file 5 with lambda and final concentrations. (See Theory, equations (3), (4), and (5)).

OUTPUT. This subprogram outputs the contents of file 5 when concentrations of all isotopes are desired. Options for printing individual concentrations, concentrations by mass number, or concentrations by atomic number are selected here.

SPCASE. This program outputs concentration of selected isotopes. With proper option selection, this routine will output as many isotopes as desired by the user.

HEAT. This subprogram calculates and outputs the individual and total average beta energy given off in one second by the fission products predicted. Energy of beta decay per isotope is input with subprogram DCYPRP (See Theory, equations (11), and (12)).

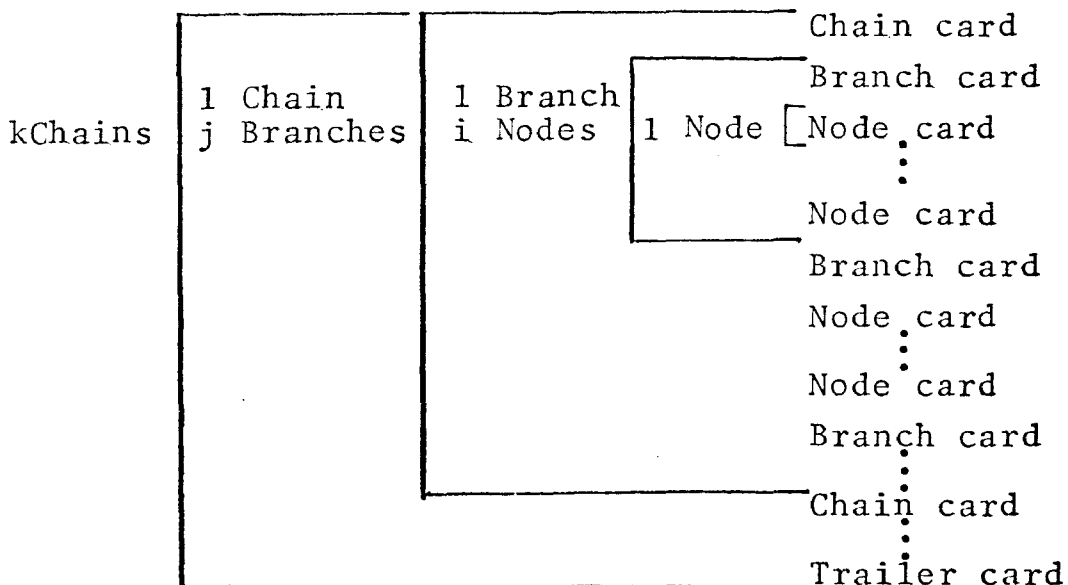
SHIELD. This subprogram inputs gamma groups by energy, scans file 5 and outputs the number of atoms decayed in one second per energy group. Energy of decay is determined and filed by subprogram DCYPRP. (See Theory, equation (13)).

ENDZAP. This routine writes all work files as they stand to tape or other specified device and frees the space of work files to the computer.

Support Utilities

DATA1. This routine reads the straight line decay chains and their associated data and records this data in more compact form on file 4. From this information, it also sets the format of file 5 with names, atomic numbers, and mass numbers. From other input data, it also writes file 18 with highest, lowest, and most stable atomic numbers per chain for all chains. This routine completes files 4 and 18.

The input to DATA1 consists of chain, branch, and node cards in a particular sequence. The branching fraction (r_m) indicated on node cards must be calculated as shown in Appendix 4. The chain number is the mass number of the first element in the decay chain.



Data Cards

<u>Information</u>	<u>Columns</u>	<u>Format</u>
<u>Chain card</u>		
Chain number	1-3	I3
Number of branches in chains	10-11	I2
<u>Branch card</u>		
Chain number	1-3	I3
Branch number	10-11	I2
Number of nodes in branch	21-22	I2
<u>Node Card</u>		
Chain number	1-3	I3
Branch number	10-11	I2
Node number	21-22	I2
Decay constant	48-61	E14.8
Branching fraction (r_m)	62-69	F8.6
Element symbol	71-72	A2
Element mass number	74-76	I3
Element state	77	A1
Atomic number	78-80	I3
<u>Trailer card</u>		
999	1-3	I3

DCYPRP. This routine calculates average beta and gamma decay per isotope (See Theory, equations (11), (12), (13), (14), (15)) and records this information on file 5 completing file 5 input.

DATA1X. This routine reads yields per mass number from cards and records the information completing files 14 through 17.

CRSPRP. This routine reads microscopic thermal cross sections from cards and records this information on file 12.

This completes file input to entire program. Options available in ZAP are explained in Appendix 3. The rest of the subprograms listed print files so that they may be checked for accuracy.

PYLD. This routine prints the contents of files 14, through 17, as prepared by DATA1X.

PXSEC. This routine prints the contents of file 12 as prepared by CRSPRP.

PRT4. This routine prints the contents of file 4 as prepared by DATA1 and DCYPRP.

F4BUG. This routine places a number as the first position of a specified record in file 4 thereby shortening the input for tests runs or checking validity of routines.

CHKZAP. This routine prints the contents of files 9 and 11 as prepared by ZAP.

Each of the above routines may have to be modified (i.e., limits of DO loops) to obtain desired results. See Appendix 5 for file formats.

APPENDIX 2
Program Listing

SUBROUTINES AND PROGRAMS MAY BE DELIMITED BY A /*
 IF THIS PROGRAM IS TO BE COPIED FROM THE LISTING, CARE
 MUST BE TAKEN IN THE TRANSLATION. ONLY 60 CHARACTERS
 ARE PERMITTED ON A LINE IN THIS THESIS. THEREFORE,
 WHEN AN 'X' APPEARS IN COLUMN 6, IT IS THE
 CONTINUATION OF THE PREVIOUS CARD BEGINNING IN COLUMN 61

THIS SECTION IS NOT A PART OF ZAP

/*

```

C   THIS IS PROGRAM STARTZAP
      DIMENSION CH(80),NNN(15),III(3),BG(4),N5(5),Z(45),XS(4
X2,109,2),
      1YLD(180)
      DEFINE FILE 4(2400,17,U,I4)
      DEFINE FILE 5(1500,45,U,I5)
      READ(1,101) IBGN,IEND
1   READ(8,100,END=10) CH
      WRITE(3,102) CH
      GO TO 1
10  CONTINUE
      WRITE(3,150)
11  READ(8,100,END=21) CH
      IF(2.LT.IBGN.OR.2.GT.IEND) GO TO 20
      WRITE(3,102) CH
20  CONTINUE
      GO TO 11
21  WRITE(3,151)
22  READ(8,100,END=28) CH
      IF(3.LT.IBGN.OR.3.GT.IEND) GO TO 25
      WRITE(3,102) CH
25  CONTINUE
      GO TO 22
28  WRITE(3,152)
      I4=1
      WRITE(3,175)
26  READ(8, ,END=36) MSS,MAXB,NNN
      IF(4.LT.IBGN.OR.4.GT.IEND) GO TO 35
      WRITE(4,I4) MSS,MAXB,NNN
      WRITE(3,105) MSS,MAXB,NNN
      IF(MSS.EQ.999)GO TO 35
      M=0
      DO 27 I=1,MAXB
27  M=M+NNN(I)
      DO 29 I=1,M
      READ(8, ) III,BG,N1,N2,N3,N4,N5
      WRITE(4,I4) III,BG,N1,N2,N3,N4,N5
      WRITE(3,106) III,BG,N1,N2,N3,N4
29  CONTINUE
35  CONTINUE
      GO TO 26
36  WRITE(3,153)
      I5=1
      WRITE(3,176)

```

```

      DO 37 I=1,1051
      READ(8,END=41) Q
      IF(5.LT.IBGN.OR.5.GT.IEND) GO TO 40
      WRITE(5,15) Q
      WRITE(3,108) Q
40  CONTINUE
37  CONTINUE
41  WRITE(3,154)
      WRITE(3,177)
      DO 44 L=1,2
      DO 44 I=1,42
      READ(8      ) (XS(I,J,L),J=1,109)
      IF(6.LT.IBGN.OR.6.GT.IEND) GO TO 45
      WRITE(12)      (XS(I,J,L),J=1,109)
      WRITE(3,110)(XS(I,J,L),J=1,109)
45  CONTINUE
44  CONTINUE
406 READ(8      ,END=407) CHAR
      GO TO 406
407 CONTINUE
      WRITE(3,155)
      DO 51 KK=7,10
      KKK=KK+7
      WRITE(3,178) KKK
      READ(8      ) YLD
      IF(KK.LT.IBGN.OR.KK.GT.IEND) GO TO 50
      WRITE(3,110) YLD
      WRITE(KKK) YLD
50  CONTINUE
408 READ(8      ,END=409) CHAR
      GO TO 408
409 CONTINUE
      WRITE(3,156) KKK
51  CONTINUE
      IF(11.LT.IBGN.OR.11.GT.IEND) GO TO 60
      WRITE(3,179)
      WRITE(3,111)
      DO 53 I=1,2
      READ(8      ) YLD
      WRITE(3,113) I
      WRITE(3,114) YLD
53  WRITE(18) YLD
      I=3
      READ(8      ) YLD
      WRITE(3,113) I
      WRITE(3,115) YLD
      WRITE(18) YLD
      WRITE(3,157)
60  WRITE(3,158)
      CALL EXIT
101 FORMAT(T15,I2,T30,I2)
102 FORMAT(T2,80A1)

```

```

100 FORMAT(80A1)
103 FORMAT(17I4)
104 FORMAT(3I4,4E10.4,2A4,2I4,6I2)
105 FORMAT(1X,17(I4,3X))
106 FORMAT(1X,3I4,4E10.4,2A4,2I4)
107 FORMAT(I4,A4,I4,A4,12E10.4,2I4,A4,12E10.4,2I4,12E10.4)
108 FORMAT(1X,I4,A4,I4,A4/12E12.4/2I4,A4/12E12.4/2I4/12E12
X.4)
109 FORMAT(40(5E12.4))
110 FORMAT(10E12.4)
111 FORMAT('1PART 1 CONTAINS ZLOW-180 EACH**PART 2 CONTAIN
XS ZHI-180 EA
1CH**PART 3 CONTAINS ZSTABLE-180 EACH')
112 FORMAT(60(30I4))
113 FORMAT('1THIS IS PART',I2//)
114 FORMAT(30I4)
115 FORMAT(10F12.4)
150 FORMAT('//T10,'END FILE FT08FC01'/'1')
151 FORMAT('//T10,'END FILE FT08FC02'/'1')
152 FORMAT('//T10,'END FILE FT08FC03'/'1')
153 FORMAT('//T10,'END FILE FT08FC04')
154 FORMAT('//T10,'END FILE FT08FC05')
155 FORMAT('//T10,'END FILE FT08FC06')
156 FORMAT('//T10,'END FILE FT08FC0',T24,I3)
157 FORMAT('//T10,'END FILE FT08FC11')
158 FORMAT('1',T10,'END OF STARTZAP')
175 FORMAT('1',T5,'THIS IS PROGRAM FILE 4'//)
176 FORMAT('1',T5,'THIS IS PROGRAM FILE 5'//)
177 FORMAT('1',T5,'THIS IS PROGRAM FILE 12'//)
179 FORMAT('1',T5,'THIS IS PROGRAM FILE 18'//)
178 FORMAT('1',T5,'THIS IS PROGRAM FILE',I3,//)
END

```

/*


```

C   THIS IS PROGRAM ZAP
      DIMENSION IHLDC(11),PGTITL(8),CLTM(22),COOLTM(11),BURN
      XTM(11),
      1BTM(22),IC(11),IS(11)
      INTEGER L/'L'/,BLK/' '/,DOL/'$'/,Y/'Y'/,I/'I'/,P/'P'/,
      XN/'N'/,
      2O/'O'/,DEC/'.'/,W/'W'/,F/'F'/,LP/'('/,EQ/'='/,A/'A'/,C
      XOM/'','/,
      3E/'E'/,CC/'C'/,B/'B'/,M/'M'/,COL/':'/,T/'T'/,S/'S'/,G/
      X'G'/,C(80)
      DATA IER,IYLD,IProg,IOBL,IMU/5*0/,CLTM/22*' '/,COOLTM/
      X11*0./,
      1LCMAX,LBMAX,IPOUT,ICASE,ICASEN,IHEAT,ICOOLEM,NBCDS/8*0/
      X,IS/11*0/,
      2ISHLD,ISMAX,IGPMAX/3*0/,BURNM/11*0./,BTM/22*' '/,IC/1
      X1*0/
      WRITE(3,120)
      WRITE(3,121)
      WRITE(3,122)
      CALL REREAD
      READ(1,100) C
      WRITE(3,101) C
      IF(C(1).NE.DOL) CALL ERR(1,C,&990,IER)
      IF(C(2).NE.DOL) CALL ERR(2,C,&990,IER)
      WRITE(3,103)
10  READ(1,100) C
      WRITE(3,101) C
C   CHECK FOR COMMENT CARDS
C
      IF(C(1).EQ.DOL) GO TO 10
C
C   FIND PROPER IYLD
      IF(C(1).NE.Y) CALL ERR(1,C,&500,IER)
      IF(C(12).EQ.I) GO TO 501
      IF(C(12).NE.P) CALL ERR(12,C,&502,IER)
      IF(C(33).EQ.I) GO TO 503
      IF(C(33).NE.P) CALL ERR(33,C,&504,IER)
      READ(99,150) IYLD
      WRITE(3,107) IYLD
15  READ(1,100) C
      WRITE(3,101) C
      IF(C(1).EQ.DOL) GO TO 15
C
C   CHECK AND SET IProg AND IOBL
      IF(C(1).NE.Y) CALL ERR(1,C,&505,IER)
      IF(C(15).NE.N) GO TO 16
      IProg=0
      IOBL=0
      GO TO 25
16  IF(C(15).NE.P) CALL ERR(15,C,&506,IER)
      IF(C(21).EQ.O) GO TO 507
      IF(C(21).NE.P) CALL ERR(21,C,&508,IER)

```

```

      IPRDG=1
      IF(C(39).EQ.BLK) GO TO 509
      IF(C(39).EQ.O ) GO TO 510
      WRITE(3,104) (C(II),II=35,42)
25  WRITE(3,113) (C(KK),KK=1,70)
      IPL=IYLD+2
C    DECIDE WHERE ACCORDING TO IYLD
C
      GO TO(27,27,36,36,36,36),IPL
27  CONTINUE
28  READ(1,100) C
      WRITE(3,101) C
      IF(C(1).EQ.DOL) GO TO 28
      WRITE(9,100) C
      IF(IPL.EQ.1) GO TO 32
208 READ(1,100) C
      WRITE(3,101) C
      IF(C(1).EQ.DOL) GO TO 208
C
C    CHECK COL 5 FOR FOR L
      IF(C(5).NE.L) GO TO 29
      IF(C(41).NE.DEC) CALL ERR(41,C,&511,IER)
      IF(C(44).EQ.BLK) CALL ERR(44,C,&511,IER)
      WRITE(9,100) C
      GO TO 31
29  IF(C(5).NE.F) CALL ERR(5,C,&512,IER)
      WRITE(9,100) C
209 READ(1,100) C
      WRITE(3,101) C
      IF(C(1).EQ.DOL) GO TO 209
      IF(C(14).NE.LP) CALL ERR(14,C,&513,IER)
      IF(C(16).NE.EQ) CALL ERR(16,C,&514,IER)
      IF(C(22).NE.EQ) CALL ERR(22,C,&515,IER)
      IF(C(49).NE.EQ) CALL ERR(49,C,&516,IER)
      WRITE(9,100) C
31  WRITE(3,113) (C(KK),KK=1,80)
      WRITE(3,108)
      DO 33 II=1,18
303 READ(1,100) C
      WRITE(3,101) C
      IF(C(1).EQ.DOL) GO TO 303
      WRITE(9,100) C
      IF(C(78).EQ.E) GO TO 306
33  CONTINUE
      GO TO 36
306 IF(II.EQ.18) GO TO 36
      WRITE(3,109)
      IER=IER+1
      GO TO 36
32  DO 34 II=1,10000
      READ(1,100) C
      WRITE(3,101) C

```

```

      IF(C(1).EQ.DOL) GO TO 34
      IF(C(1).NE.A) CALL ERR(1,0,&517,IER)
      IF(C(8).NE.FQ) CALL ERR(8,0,&518,IER)
      IF(C(20).NE.EQ) CALL ERR(20,0,&519,IER)
      IF(C(30).NE.EQ) CALL ERR(30,0,&520,IER)
      WRITE(9,100) C
      IF(C(45).EQ.COM) GO TO 36
34  CONTINUE

C
C      END DTPAR INPUT - BEGIN CROSS INPUT
C

36  READ(1,100) C
      WRITE(3,101) C
      IF(C(1).EQ.DOL) GO TO 36
      IF(C(3).NE.EQ) CALL ERR(3,0,&521,IER)
      IF(C(4).EQ.D) GO TO 522
      IF(C(4).EQ.L) GO TO 523
      IF(C(4).NE.P) CALL ERR(4,0,&524,IER)
      IMU=3
      WRITE(3,112)
37  READ(1,100) C
      WRITE(3,101) C
      IF(C(1).EQ.DOL) GO TO 37
      IF(C(1).NE.W.AND.C(1).NE.I) CALL ERR(1,0,&525,IER)
      WRITE(9,100) C
      WRITE(3,114) (C(KK),KK=1,25)
C      CHECK PHI CARD
38  READ(1,100) C
      WRITE(3,101) C
      IF(C(1).EQ.DOL) GO TO 38
      IF(C(1).NE.P) CALL ERR(1,0,&526,IER)
      IF(C(4).NE.EQ) CALL ERR(4,0,&526,IER)
      IF(C(63).NE.LP.AND.C(63).NE.BLK) CALL ERR(63,0,&527,IE
XR)
      WRITE(9,100) C
      WRITE(3,113) (C(KK),KK=1,18),(C(KK),KK=31,38),(C(KK),K
XK=62,69)
C      WRITE CROSS SECTIONS TO DISK
401 WRITE(3,127)
      DO 40 II=1,10000
39  READ(1,100) C
      WRITE(3,101) C
      IF(C(1).EQ.DOL) GO TO 39
      IF(C(2).NE.EQ) CALL ERR(2,0,&40,IER)
      IF(C(8).NE.EQ) CALL ERR(8,0,&40,IER)
      IF(C(15).NE.EQ) CALL ERR(15,0,&40,IER)
      IF(C(35).NE.EQ.AND.C(35).NE.BLK) CALL ERR(35,0,&40,IE
XR)
      IF(C(55).NE.EQ.AND.C(55).NE.BLK) CALL ERR(55,0,&40,IE
XR)
      WRITE(9,100) C
      IF(C(72).EQ.E) GO TO 41

```

```

40 CONTINUE
C
C   END CROSS INPUT - BEGIN OUTPUT INPUT
C
41 READ(1,100) C
   WRITE(3,101) C
   IF(C(1).EQ.DOL) GO TO 41
   READ(99,151) PGTITL
   WRITE(3,123) PGTITL
C   READ CLTM,COOLMX,LCMAX
   DO 45 II=1,11
42 READ(1,100) C
   WRITE(3,101) C
   IF(C(1).EQ.DOL) GO TO 42
   IF(C(10).NE.CC) CALL ERR(10,0,&528,IER)
   READ(99,152) CLTM(2*II-1),CLTM(2*II),IHLDC(II)
   READ(99,153) COOLTM(II)
   WRITE(3,124) C(23),C(24),C(9),(C(KK),KK=1,8)
   LCMAX=II
   IF(C(25).EQ.COM) GO TO 47
45 CONTINUE
   IER=IER+1
   WRITE(3,115)
47 DO 50 II=1,11
48 READ(1,100) C
   WRITE(3,101) C
   IF(C(1).EQ.DOL) GO TO 48
   IF(C(10).NE.I) CALL ERR(10,0,&529,IER)
   READ(99,152) BTM(2*II-1),BTM(2*II)
   READ(99,153) BURNTM(II)
   WRITE(3,125) C(27),C(28),C(9),(C(KK),KK=1,8)
   LBMAX=II
   IF(C(29).EQ.COM) GO TO 52
50 CONTINUE
   IER=IER+1
   WRITE(3,116)
C   END READ BURNTM,BTM,LBMAX
C   READ OUTPUT FORMAT
52 READ(1,100) C
   WRITE(3,101) C
   IF(C(1).EQ.DOL) GO TO 52
   IF(C(1).NE.O) GO TO 60
   IF(C(16).NE.T) GO TO 53
   IPOUT=3
51 WRITE(3,113)(C(KK),KK=1,40)
   GO TO 55
53 IF(C(15).NE.M) GO TO 54
   IPOUT=2
   GO TO 51
54 IF(C(16).NE.L) CALL ERR(14,16,&530,IER)
   IPOUT=1
   GO TO 51

```

```

C   READ OPTION
55  READ(1,100) C
    WRITE(3,101) C
    IF(C(1).EQ.DOL) GO TO 55
    ICASE=0
    IF(C(7).NE.COL) GO TO 65
    ICASE=1
    WRITE(3,113) (C(KK),KK=1,60)
    GO TO 602
60  IF(C(1).NE.N) CALL ERR(1,0,&531,IER)
    WRITE(3,113)(C(KK),KK=1,78)
    IF(C(11).NE.BLK) GO TO 62
    ICASE=0
    IPUT=0
    GO TO 64
62  IF(C(11).NE.L) CALL ERR(11,0,&532,IER)
    ICASE=1
    IPUT=0
602 DO 63 II=1,10000
603 READ(1,100) C
    WRITE(3,101) C
    IF(C(1).EQ.DOL) GO TO 603
    WRITE(9,100) C
    ICASEN=II
    IF(C(27).EQ.COM) GO TO 64
63  CONTINUE
64  READ(1,100) C
    WRITE(3,101) C
    IF(C(1).EQ.DOL) GO TO 64
C   END OUTPUT INPUT - BEGIN HEAT INPUT
65  CONTINUE
66  IF(C(1).NE.N) GO TO 68
    IHEAT=0
    GO TO 545
68  IF(C(1).NE.CC) GO TO 74
    IHEAT=1
    GO TO 546
609 IF(C(38).NE.EQ) CALL ERR(38,0,&533,IER)
    IKP=0
    IF(C(39).EQ.A) GO TO 534
    DO 70 II=1,11
    IF(C(38+3*II).NE.COM) GO TO 71
    IKP=IKP+1
70  CONTINUE
71  ICOOLM=IKP+1
    READ(99,154) (IC(II),II=1,ICOOLM)
    DO 72 III=1,ICOOLM
    DO 72 II=1,LCMAX
    IF(IC(III).EQ.IHLDC(II)) IC(III)=II
72  CONTINUE
    GO TO 80
74  IF(C(1).NE.S) CALL ERR(1,0,&535,IER)

```

```

      GO TO 547
740 IF(C(38).NE.EQ) CALL ERR(38,C,&533,IER)
      IKP=0
      IF(C(39).EQ.A) GO TO 536
      DO 75 II=1,11
      IF(C(38+3*II).NE.COM) GO TO 76
      IKP=IKP+1
75 CONTINUE
76 ICCOLM=IKP+1
      READ(99,154) (IC(II),II=1,ICCOLM)
      DO 79 III=1,ICCOLM
      DO 79 II=1,LCMAX
      IF(IC(III).EQ.IHLDC(II)) IC(III)=II
79 CONTINUE
      NBCDS=0
77 READ(1,100) C
      WRITE(3,101) C
      IF(C(1).EQ.DOL) GO TO 77
      IF(C(1).NE.A) CALL ERR(1,C,&535,IER)
      IF(C(11).NE.EQ) CALL ERR(11,C,&561,IER)
      IF(C(19).NE.EQ) CALL ERR(19,C,&562,IER)
      WRITE(9,100) C
      NBCDS=NBCDS+1
880 IF(C(23).EQ.COM) GO TO 80
      GO TO 77

```

C
C
C

END HEAT INPUT - BEGIN SHIELD INPUT

```

80 READ(1,100) C
      WRITE(3,101) C
      IF(C(1).EQ.DOL) GO TO 80
      IF(C(1).NE.N) GO TO 81
      ISHLD=0
      WRITE(3,113) (C(KK),KK=1,60)
      GO TO 89
81 IF(C(1).NE.G) CALL ERR(1,C,&538,IER)
      ISHLD=1
      GO TO 544
82 IF(C(35).NE.EQ) CALL ERR(35,C,&538,IER)
      IF(C(36).EQ.A) GO TO 539
      ISMAX=0
      DO 84 II=1,11
      IF(C(35+3*II).NE.COM) GO TO 85
      ISMAX=ISMAX+1
84 CONTINUE
85 ISMAX=ISMAX+1
      READ(99,156) (IS(II),II=1,ISMAX)
      DO 860 III=1,ISMAX
      DO 860 II=1,LCMAX
      IF(IS(III).EQ.IHLDC(II)) IS(III)=II
860 CONTINUE
86 IGPMAX=0

```

```

87 READ(1,100) C
   WRITE(3,101) C
   IF(C(1).EQ.DOL) GO TO 87
   IF(C(1).NE.G) CALL ERR(1,0,&540,IER)
   IF(C(10).NE.EQ) CALL ERR(10,C,&541,IER)
   IF(C(25).NE.EQ) CALL ERR(25,C,&542,IER)
   IF(C(47).NE.EQ) CALL ERR(47,C,&543,IER)
   WRITE(9,100) C
   IGPMAX=IGPMAX+1
88 IF(C(57).EQ.COM) GO TO 89
   GO TO 87
89 READ(1,100,END=94) C
   WRITE(3,101) C
   IF(C(1).NE.DOL) CALL ERR(1,0,&548,IER)
   IF(C(2).NE.DOL) GO TO 89
   WRITE(3,126)
94 ENDFILE 9
   WRITE(11) IER,IYLD,IPROG,IOBL,IMU,PGITTL,CLTM,COULTM,L
   XCMAx,
   ABURNM,BTM,LBMAX,IPOUT,ICASE,ICASEN,IHEAT,ICODLM,IC,NB
   XCDS,ISHLD,
   BIS,ISMAX,IGPMAX
   ENDFILE 11
   IF(IER.EQ.0) GO TO 98
   WRITE(3,118) IER
   CALL EXIT
98 WRITE(3,119)
   CALL EXIT
990 WRITE(3,102)
   CALL EXIT
500 WRITE(3,104) (C(II),II=1,5)
   GO TO 15
501 IYLD=-1
   WRITE(3,105)
   GO TO 15
502 WRITE(3,104) (C(II),II=9,15)
   GO TO 15
503 WRITE(3,106)
   IYLD=0
   GO TO 15
504 WRITE(3,104) (C(II),II=30,35)
   GO TO 15
505 WRITE(3,104) (C(II),II=1,5)
   GO TO 25
506 WRITE(3,104) (C(II),II=11,18)
   GO TO 25
507 IPROG=0
   IOBL=1
   GO TO 25
508 WRITE(3,104) (C(II),II=17,24)
   GO TO 25
509 IOBL=0

```

```

      GO TO 25
510 IOBL=1
      GO TO 25
511 WRITE(3,104) (C(II),II=39,48)
      GO TO 209
512 WRITE(3,104) (C(II),II=2,8)
      GO TO 209
513 WRITE(3,104) (C(II),II=11,17)
      GO TO 31
514 WRITE(3,104) (C(II),II=14,20)
      GO TO 31
515 WRITE(3,104) (C(II),II=19,25)
      GO TO 31
516 WRITE(3,104) (C(II),II=46,54)
      GO TO 31
517 WRITE(3,104) (C(KK),KK=1,5)
      GO TO 34
518 WRITE(3,104) (C(KK),KK=5,13)
      GO TO 34
519 WRITE(3,104) (C(KK),KK=17,24)
      GO TO 34
520 WRITE(3,104) (C(KK),KK=27,34)
      GO TO 34
521 WRITE(3,104) (C(KK),KK=1,6)
      GO TO 37
522 IMU=2
      WRITE(3,111)
      GO TO 41
523 IMU=1
      WRITE(3,110)
      GO TO 41
524 WRITE(3,104) (C(KK),KK=1,7)
      GO TO 37
525 WRITE(3,104) (C(KK),KK=1,6)
      GO TO 38
526 WRITE(3,104) (C(KK),KK=1,9)
      GO TO 401
527 WRITE(3,104) (C(KK),KK=60,66)
      GO TO 401
528 WRITE(3,104) (C(KK),KK=7,14)
      GO TO 45
529 WRITE(3,104) (C(KK),KK=7,14)
      GO TO 50
530 WRITE(3,104) (C(KK),KK=13,18)
      GO TO 55
531 WRITE(3,104) (C(KK),KK=1,7)
      GO TO 55
532 WRITE(3,104) (C(KK),KK=7,14)
      GO TO 64
533 WRITE(3,104) (C(KK),KK=35,42)
      GO TO 80
534 DO 604 III=1,LCMAX

```



```

604 IC(III)=II
    ICOCLY=LMAX
    GO TO 80
535 WRITE(3,104) (C(KK),KK=1,7)
    GO TO 880
536 ICOCLY=LMAX
    DO 605 II=1,LMAX
605 IC(II)=II
    GO TO 77
537 WRITE(3,113) (C(KK),KK=1,35)
    GO TO 66
538 WRITE(3,104) (C(KK),KK=1,7)
    GO TO 86
539 ISMAX=LMAX
    DO 801 II=1,LMAX
801 IS(II)=II
    GO TO 86
540 WRITE(3,104) (C(KK),KK=1,7)
    GO TO 88
541 WRITE(3,104) (C(KK),KK=7,15)
    GO TO 88
542 WRITE(3,104) (C(KK),KK=22,29)
    GO TO 88
543 WRITE(3,104) (C(KK),KK=44,51)
    GO TO 88
544 WRITE(3,113) (C(KK),KK=1,80)
    GO TO 82
545 WRITE(3,113) (C(KK),KK=1,80)
    GO TO 80
546 WRITE(3,113) (C(KK),KK=1,80)
    GO TO 609
547 WRITE(3,113) (C(KK),KK=1,80)
    IHEAT=2
    GO TO 740
548 WRITE(3,117)
    GO TO 89
561 WRITE(3,104) (C(KK),KK=8,15)
    GO TO 880
562 WRITE(3,104) (C(KK),KK=16,22)
    GO TO 880

```

C
C
C

```

    FORMAT STMTS
100 FORMAT(80A1)
101 FORMAT(T41,80A1)
102 FORMAT(/T40,'SUPERZAP ENTRY CARD MISPLACED-JOB TERMIN
    XATED')
103 FORMAT(T2,'*** BEGIN SUPERZAP INPUT ***')
104 FORMAT(T2,'**MISPLACED CHARACTER IN...',10A1,'CARD SKIP
    XPED**')
105 FORMAT(T2,'# PARTIAL YIELD INPUT CHOSEN')
106 FORMAT(T2,'# USER YIELD CURVE CHOSEN-PROGRAM DISTRIBUT
    XION CHOSEN')

```

```

107 FORMAT(T2,'# YIELD DATA TAKEN FROM PROGRAM CURVE',I2)
108 FORMAT(T2,'*** WILL NOT CHECK POSITIONS OF YIELD CHARA
XCTERS***')
109 FORMAT(T2,'**IMPROPER NO. OF INPUT YIELD DATA CARDS**'
X)
110 FORMAT(T2,'# SET MU=LAMBDA')
111 FORMAT(T2,'# LEAVE MU UNCHANGED')
112 FORMAT(T2,'# SET MU=PHI*SIGMA+LAMBDA')
113 FORMAT(T2,'# ',80A1)
114 FORMAT(T2,'# ',25A1)
115 FORMAT(T2,'**TOO MANY COOLING TIMES OR '',END'' MISSIN
XG**')
116 FORMAT(T2,'**TOO MANY BURNUP TIMES OR '',END'' MISSING
X**')
117 FORMAT(T2,'**TOO MANY CARDS IN DECK**')
118 FORMAT(/T22,'***THIS DATA CHECK HAS ',I4,' TERMINAL E
XRRORS-DATA M
CAY OR MAY NOT BE ON FILE IN PROPER ORDER***'/T45,'****
X**SUGGEST NO
DGO STATUS TO SUPERZAP*****')
119 FORMAT(/T40,'**NO ERRORS DETECTED-SUGGEST GO STATUS
XTO SUPERZAP*
J**')
120 FORMAT('1',T51,'PROGRAM=ZAP INPUT DATA CHECK MADE'/T
X51,'LEGEND: '
E/T51,'*** DENOTES INFORMATION'/T51,'** DENOTES ERROR
XCONDITION MA
FRKED BY '$'/'/T51,'# DENOTES PROGRAM SETTINGS')
121 FORMAT(/T33,'***FIRST ERROR DETECTED IS CERTAIN-OTHER
XERRORS MAY R
RESULT FROM THIS***')
122 FORMAT(/T75,'CARD COLUMNS'/T41,'000000000111111111122
X222222223333
G333333344444444445555555555666666666677777777778'/T41,'
X123456789012
H345678901234567890123456789012345678901234567890123456
X789012345678
I90'/)
123 FORMAT(T2,'# PAGE TITLE IS ',8A4,'')
124 FORMAT(T2,'# COOLING TIME NO. ',14A1)
125 FORMAT(T2,'# IRRADIATION TIME NO. ',14A1)
126 FORMAT(T2,'*** END SUPERZAP INPUT ***')
127 FORMAT(T2,'***CROSS SECTION DATA-ERRORS MARKED WITH $
XONLY***')
150 FORMAT(T52,I2)
151 FORMAT(8A4)
152 FORMAT(2A4,T23,I2)
153 FORMAT(F4.1)
154 FORMAT(T39,11(I2,1X))
155 FORMAT(T38,11(I2,1X))
156 FORMAT(T36,11(I2,1X))
END

```

/*

```

SUBROUTINE ERR(I,J,*,IER)
INTEGER C(80),DOL,BLK
DATA C/80*' ','/','DOL/','$','/','BLK/',' '
DO 30 K=I,J
30 C(K)=DOL
WRITE(3,100) C
DO 35 K=I,J
35 C(K)=BLK
IER=IER+1
RETURN 1
100 FORMAT(T41,80A1)
END

```

/*

```

REREAD$ CSECT
REREAD  ENTER
      L      1,=A(ZZZ)
      L      15,=V(IHCFOMH)
      MVI    74(15),X'50'
      EX     0,74(15)
      MVI    74(15),X'58'
      RETRN
      SPACE  3
      USING  *,1
ZZZ    STM    4,15,SAVE
      DROP   1
      USING  ZZZ,11
      LR     11,1
      LR     4,0
      CLC    0(2,4),INIT      GET ADDR. OF TYPE OF CALL IND
                                IS THIS A READ INITIALIZATION
      X?
      BE     CHK99            B IF YES
      L      1,=V(IHCFIOH)
      LR     0,4
      LM     4,15,SAVE
      BR     1
CHK99  CLC    0(4,14),=F'99' (NO EOF, NO ERR, UNIT= CONSTAN
XT "99")
      BE     ITIS99            B IF DSRN = 99
      LR     2,14              PUT DSRN OR A(DSRN) INTO 2
      L      1,=V(IHCFIOH)
      BALR   0,1
INIT   DC     X'00F0'          * SIMULATE THE CALL FROM IBCOM
      STM    2,3,SV1          * TO FIOCS= FOR INIT.
                                STORE BUFFER ADDRESS & BUFFER
      XLENGTH
      B      LEAVE
      SPACE  3
ITIS99 LM     2,3,SV1          RESTORE 2 & 3 AS THEY WERE AFT
XER
      SPACE  1
LEAVE  LA     1,2(4)          GET RETURN ADDR.
      LM     4,15,SAVE

```

```

        BR      1
        EJECT
SAVE    DC      18F'0'
SV1     DS      F
SV2     DS      F
        LTORG
        END

```

```
/*
```

```
MEMBER NAME ENTER
```

```
        MACRO
&NAME   ENTER &BASE=12
        LCLC   &C1

```

```
&C1     SETC   'SVR&SYSNDX'
```

```
        AIF    ('&BASE' GT '2' AND '&BASE' LT '13').BASECK
```

```
        MNOTE  13,'REGISTER &BASE MAY NOT BE USED FOR A BASE
X REG.'
```

```
        MEXIT

```

```
.BASECK ANOP
```

```
        MNOTE *,' &BASE IS THE BASE REGISTER FOR THIS ROUTINE'
```

```
        MNOTE *,' &C1 IS THE 18 WORD SAVE AREA FOR THIS ROUTINE'
```

```
        MNOTE *,' &NAME IS THE LOGICAL ENTRY POINT FOR THIS ROUTINE'
```

```
        XE'
```

```

        ENTRY &NAME           DEFINE ENTRY POINT
        USING *,&BASE         SET UP BASE REGISTERING
&NAME   SAVE   (14,12),,*     SAVE REGISTERS 14,15,0,1,...,1
        X2
        LR      &BASE,15      PROVIDE ADDRESSABILITY
        LA      2,&C1         PICK UP ADDR OF THIS SAVE AREA
        ST      2,8(0,13)     STORE A(SAVE) IN CALLER'S SAVE
        X AREA.
        ST      13,4(0,2)     STORE CALLER'S SAVE AREA ADDR
        XIN SAVE
        LR      13,2          SET UP SAVE AREA POINTER REG.
        B       BYP&SYSNDX    BOUNCE AROUND SAVE AREA
        DS      0D
&C1     DS      18F          18 WORD SAVE AREA
BYP&SYSNDX EQU *
        MEND

```

```
/*
```

```
MEMBER NAME RETRN
```

```
        MACRO
&NAME   RETRN &RC=
&NAME   L      13,4(0,13)

```

```
PICK UP THE OLD SAVE AREA ADDR
```

```
        X.
```

```
        LM      14,15,12(13)  RESTORE THE REGS. 14,15
```

```
        LM      2,12,28(13)   RESTORE THE REGS. 2,3,...,12
```

```
        MVI     12(13),X'FF'  SET RETURN INDICATION
```

```
        AIF     (T'&RC EQ '0').NORC
```

```
        AIF     ('&RC' LT '4096').PCOK
```

```
        MNOTE  1,'RETURN CODE RC=&RC IS > 4095, RC IGNORED'
```

```
        AGO     .NORC
```

```
.RCOK   LA      15,&RC.(0,0)  LOAD RETURN CODE
```

.NORC

BR

14

RETURN

MEND

/*

```

DATA SEC,MIN,HR,DAY,YR/' SEC',' MIN',' HRS',' DAYS',' Y
XRS'/
C BTM=ALPHA BURNUP TIMES
C CLTM= ALPHA COOLING TIMES
C LBMAX=MAX BURNUP TIMES
C LCMAX= MAX COOLING TIMES
C BURNTM=RETURNED BURNUP TIMES IN SEC/INITIAL AS READ
C COOLTM=RETURNED COOLING TIMES IN SEC/INITIAL AS READ
  TDAY=3600.*24.
  TYR=TDAY*365.25
  DO 10 I=1,LCMAX
    A=CLTM(2*I)
    IF(A.EQ.YR) COOLTM(I)=COOLTM(I)*TYR
    IF(A.EQ.DAY) COOLTM(I)=COOLTM(I)*TDAY
    IF(A.EQ.HR) COOLTM(I)=COOLTM(I)*3600.
    IF(A.EQ.MIN) COOLTM(I)=COOLTM(I)*60.
  10 CONTINUE
  DO 20 I=1,LBMAX
    A=BTM(2*I)
    IF(A.EQ.YR) BURNTM(I)=BURNTM(I)*TYR
    IF(A.EQ.DAY) BURNTM(I)=BURNTM(I)*TDAY
    IF(A.EQ.HR) BURNTM(I)=BURNTM(I)*3600.
    IF(A.EQ.MIN) BURNTM(I)=BURNTM(I)*60.
  20 CONTINUE
  RETURN
  END

/*
SUBROUTINE DTAPRP
C
C THIS ROUTINE PREPARES RECORD TYPE B WITH MU AND YIELDS
C
COMMON PTR4,X9(49),IYLD,IOBL,X8(38),IPROG
DIMENSION Q(17),ZS(180),Y(68,180),MXN(15),YTOT(180),TI
XTLE(20)
INTEGER ALOW,AHI,ZLOW(180),ZHI(180),PTR4
REAL LAMBDA
REAL LIBR
EQUIVALENCE (Q(1),MASS),(Q(7),YLD),(Q(10),NATOM),(Q(11
X),NSTATE),
1(Q(4),LAMBDA)
DATA YTOT/180*C./,ITITLE/C/,END/'END '//,LIBR/'LIBR'/
DATA ALOW,AHI/72,161/
DO 10 I=1,68
DO 10 J=1,180
10 Y(I,J)=0.
  AHI=161
  ALOW=72
C DATA DEFINITIONS
C Q() ELEMENTS OF RECORD A EQUIVALENCED
C ZS() MOST STABLE ATOMIC NO PER MASS NO.
C ZHI() HIGHER MASS NO PER CHAIN
C ZLOW() LOWEST MASS NO PER CHAIN

```

```

C      YTOT()      TOTAL YIELD OF FISSION(ABSOLUTE) PER MASS NO.
C      Y(,)        PARTIAL YIELD OF FISSION PER AT. NO. PER MASS
C      X NO.
C      ALOW        LOWEST MASS NO. CONSIDERED
C      AHI          HIGHEST MASS NO. CONSIDERED
C      AF          COMPOUND FISSION NUCLEI MASS
C      ZF          COMPOUND FISSION NUCLEI AT. NO.
C      ALH          AF MINUS FISSION NEUTRONS PRODUCED
C      IPROG        SWITCH 1-PRINT PROGENITORS 0- NO PRINT
C      IYLD SWITCH >0 - USE LIBRARY, =0 - USE PROGRAMMERS TO
C      XTAL YIELDS
C                      .0 - USE PROGRAMMERS PARTIAL YIELDS
C      ITITLE SWITCH =0 - PRINT PROGENITOR HEADING, =0 - SK
C      XIP HEADING
C      CHECK TO SEE IF READ YIELDS FROM TAPE OR CARDS
C      IF(IYLD)500,550,14
C      14 IYLD=IYLD+13
C      READ(IYLD) AF,ZF,ALH,(YTOT(I),I=4,180)
C      15 READ(18) ZLOW
C      READ(18) ZHI
C      READ(18) ZS
C
C      BEGIN PARTIAL YIELD CALCULATIONS
C      DO 200 IAL=ALOW,AHI
C      AL=IAL
C      AH=ALH-AL
C
C      INTERPOLATE TO FIND ZHI STABLE - FIND ZLOW STABLE
C      IAH=AH
C      FIAH=IAH
C      ZLS=ZS(IAL)
C      ZHS=ZS(IAH)+(ZS(IAH+1)-ZS(IAH))*(AH-FIAH)
C      FIND ZLOW AND ZHI PROBABLE AND FINAL YIELD
C      ZLP=(ZLS-ZHS+ZF)/2.
C      ZHP=ZF-ZLP
C      YIELD=YTOT(IAL)
C      J=ZLOW(IAL)
C      K=ZHI(IAL)
C      SUM=0.
C      DO 98 IAND=J,K
C      A=IAND
C      YHOLD=YIELD*EXP(-(A-ZLP)**2)/1.772454
C      IF(YHOLD.LT.0.) YHOLD=0.
C      Y(IAND,IAL)=YHOLD
C      98 SUM=SUM+YHOLD
C      YPROG=YIELD-SUM
C      IF(YPROG.LT.0.) YPROG=0.
C      Y(J,IAL)=Y(J,IAL)+YPROG
C
C      OPTION - PRINT PROGENETORS
C      IF(IPROG.EQ.0) GO TO 200
C      IF(ITITLE.GT.0) GO TO 99

```

```

        WRITE(3,600)
        ITITLEF=1
    99 WRITE( 3 ,601)IAL,YPROG
    200 CONTINUE
    201 PTR4=1
C
C READ FILE 4 AND UPDATE RECORD A
    210 READ(4,PTR4) MASSN,MXB,MXN
        IF(MASSN.EQ.999) GO TO 400
        K=0
        DO 215 J=1,MXB
    215 K=K+MXN(J)
        DO 300 I=1,K
            READ(4,PTR4) Q
            PTR4=PTR4-1
            YLD=Y(NATOM,MASS)
            IF(IYLD.LT.0) GO TO 300
            G=NSTATE
            IF(NSTATE.GT.1) YLD=YLD/G
    300 WRITE(4,PTR4) Q
            GO TO 210
    400 PTR4=1
            IF(IOBL.NE.1) GO TO 450
            WRITE(3,609)
            DO 410 I=ALOW,AHI
                JK=ZLOW(I)
                JL=ZHI(I)
                DO 409 J=JK,JL
                    WRITE(3,610) I,J,Y(J,I)
    409 CONTINUE
    410 CONTINUE
    450 CONTINUE
        WRITE( 3 ,602)
        REWIND 14
        REWIND 18
        RETURN
    500 READ(1,605) TITLE
        WRITE(3,606) TITLE
    501 READ(1,603) NATOM,MASS,Y(NATOM,MASS),ENDCK
        WRITE(3,603) NATOM,MASS,Y(NATOM,MASS)
        IF(ENDCK.EQ.END) GO TO 201
        GO TO 501
    550 READ(1,605) TITLE
        WRITE(3,606) TITLE
        READ(1,608) (TITLE(KK),KK=1,4),IDTA
        IF(TITLE(2).EQ.LIBR) GO TO 552
        READ(1,607) IAF,IZ,AA
        AF=IAF
        ZF=IZF
        ALH=AF-AA
        GO TO 551
    552 IDTA=IDTA+13

```



```

      READ(IDTA) AF,ZF,ALH
551 READ(1,604) (YTOT(I),I=ALOW,AHI)
      GO TO 15
600 FORMAT('1',T55,'PROGENITOR OPTION EFFECTED'//)
601 FORMAT(T45,'PROCENITOR TO CHAIN ',I3,' IS ',E14.8,' AT
      XOMS')
602 FORMAT('1  END SUBROUTINE - DATA PREP')
603 FORMAT(T9,I2,T21,I3,T31,E14.8,T46,A4)
604 FORMAT(5(E14.8,2X))
605 FORMAT(20A4)
606 FORMAT(' ',20A4)
607 FORMAT(T17,I3,T23,I2,T50,F3.1)
608 FORMAT(4A4,T43,I2)
609 FORMAT('1',T54,'ORLONG TABLE OPTION EFFECTED'//)
610 FORMAT(T35,'PARTIAL YIELD TO MASS NO.',I4,' ATOMIC NO.
      X',I3,
      1' IS ',E14.8,' ATOMS')
      END
/*
      SUBROUTINE CROSS
      COMMON PTR4,PTR5,X9(50),IMU
      DIMENSION Q(17),TITLE(20),MAXN(15)
      REAL MU(42,109,2),MUL,LMDA,MS
      INTEGER PTR4,E,Z
      EQUIVALENCE (Q(1),MASS),(Q(4),LMDA),(Q(5),MUL),(Q(10)
      X,NATOM),
      1(Q(8),NAME),(Q(9),STATE)
      DATA MU/9156*0./,END/'END '//,ENDCK/' '//,LINE/0/,IPRI
      XNT/0/
      DATA TITLE/'CONV','ERT ','- LA','MBDA','=MU ',15A' '//
      DATA INDE/'INDE',//,PHI/0./,B/'B'//,MS/'M'//,Z/' '//,E/'E'//
C      DATA DEFINITIONS
C      LINE      0-PRINT HEADING,  -=0 - SKIP HEADING
C      PHI       FLUX
C      IPRI      1-PRINT MU,LAMBDA,AND/OR X-SECTIONS,  -=0 S
C      XSKIP PRINT
C      X(,)      X-SECTION HOLD AREA
C      MUL       FINAL MU
C      IMU =2-USE OLD MU  =1-MU=LAMBDA  NOT1 OR 2-FIND NEW
C      X MU
C      JXSEC     =1-PRINT X-SECTIONS,  =0 SKIP
C      JMU       =1-PRINT MU'S      ,  =0-SKIP
C      JLMDBA    =1-PRINT LAMBDA,    =0 SKIP
C
      IF(IMU.EQ.2) GO TO 1000
      IF(IMU.EQ.1) GO TO 198
      READ(1,100) TITLE
      WRITE(3,103) TITLE
      READ(1,102) PHI,JMU,JXSEC,JLMDBA
      IF(JMU.NE.E.AND.JMU.NE.Z) IPRI=1
      IF(JXSEC.NE.E.AND.JXSEC.NE.Z) IPRI=1
      IF(JLMDBA.NE.E.AND.JLMDBA.NE.Z) IPRI=1

```

```

C
C  USE INDEPENDENT CROSS SECTIONS OR WESTCOTT APPROXIMATIONS
    IF(TITLE(1).EQ.INDE) GO TO 60
C
C  READ 2200 METER CROSS SECTIONS FROM LIBRARY
    DO 2 L=1,2
    DO 2 I=1,42
    2 READ(12) (MU(I,J,L),J=1,109)
C
C  READ INPUT DATA IF S=C, R CONTAINS RS - IF RS=0, T CON
    XTAINS G+RS
C  IF MASS=0 USE SAME G+RS FOR ALL MASS NO.'S THIS AT. NO.
    4 IF(ENDCK.EQ.END) GO TO 200
    READ(1,101) NATOM,MASS,STATE,T,R,S,ENDCK
    IF(S.EQ.0.) S=1.
    A=T+R*S
    IF(MASS.EQ.0) GO TO 49
    IF(NATOM.NE.51) GO TO 20
    IF(MASS.NE.124) GO TO 20
    IF(STATE.NE.MS) GO TO 20
    L=1
    NATOM=68
    MASS=180
    GO TO 21
20  L=1
    IF((STATE.EQ.B).OR.(STATE.EQ.MS)) L=2
21  MU(NATOM-26,MASS-71,L)=MU(NATOM-26,MASS-71,L)*A
    GO TO 4
49  DO 50 K=1,109
    DO 50 L=1,2
50  MU(NATOM-26,K,L)=MU(NATOM-26,K,L)*A
    IF(NATOM.EQ.51) MU(42,109,1)=MU(42,109,1)*A
    GO TO 4
C
C  INDEPENDENT CROSS SECTION READIN SAME FORMAT AS WESTCOTT
    X S&T DUMMY
60  IF(ENDCK.EQ.END) GO TO 200
    READ(1,101) NATOM,MASS, STATE,R,S,T,ENDCK
    IF(MASS.EQ.0) GO TO 65
    IF(NATOM.NE.51) GO TO 62
    IF(MASS.NE.124) GO TO 62
    IF(STATE.NE.MS) GO TO 62
    L=1
    NATOM=68
    MASS=180
    GO TO 63
62  L=1
    IF((STATE.EQ.MS).OR.(STATE.EQ.B)) L=2
63  MU(NATOM-26,MASS-71,L)=R
    GO TO 60
65  DO 66 L=1,2
    DO 66 K=1,109

```

```

66 MU(NATOM-26,K,L)=R
   IF(NATOM.EQ.51) MU(42,109,1)=R
   GO TO 60
200 WRITE(3,104) PHI,JMU,JXSEC,JLMDA
198 PHI=PHI*1.E-24
   PTR4=1
201 READ(4,PTR4) MASS,MAXB,MAXN
   IF(MASS.EQ.999) GO TO 999
   M=0
   DO 202 I=1,MAXB
202 M=M+MAXN(I)
C READ MASS & BRANCH CARDS
C READ CHAIN NODE CARDS REPLACE MU CHECK OPTIONS
   DO 250 I=1,M
   READ(4,PTR4) Q
   PTR4=PTR4-1
   FIND(4,PTR4)
   NN=NATOM
   MM=MASS
   IF(NN.NE.51) GO TO 220
   IF(MM.NE.124) GO TO 220
   IF(STATE.NE.MS) GO TO 220
   L=1
   NN=68
   MM=180
   GO TO 222
220 L=1
   IF((STATE.EQ.MS).OR.(STATE.EQ.B)) L=2
222 MUL=MU(NN-26,MM-71,L)*PHI+LMDA
   WRITE(4,PTR4) Q
   IF(IPRINT.EQ.0) GO TO 250
   IF(LINE.NE.0) GO TO 400
   LINE=1
   WRITE(3,105)
400 WRITE(3,106) MASS,STATE,NATOM,NAME
   IF(JMU.EQ.E) GO TO 401
   WRITE(3,107) MUL
401 IF(JXSEC.EQ.E) GO TO 402
   WRITE(3,108) MU(NN-26,MM-71,L)
402 IF(JLMDA.EQ.E) GO TO 250
   WRITE(3,109) LMDA
250 CONTINUE
   GO TO 201
999 WRITE(3,103) END,TITLE
   WRITE(3,110)
1000 RETURN
100 FORMAT(20A4)
101 FORMAT(T3,I2,T9,I3,A1,T16,E14.8,T36,E14.8,T56,E14.8,T7
   X2,A4)
102 FORMAT(4X,E14.8,T64,3(A1,1X))
103 FORMAT(' ',21A4)
104 FORMAT(' 1PHI=',E14.8,',OPTIONS-PRINT(MU,X-SECTIONS,LAM
   XBDA)=(',2(A1

```

```

1,','),A1,')')
105 FORMAT(' ',T42,'OPTION TO PRINT SOME OR ALL X-SECTION
XDATA EFFECTE
1D',/T3,'ALL ISOTOPES PRINTED - 0'S UNDER MASS DENOTE
XSAME X-SECTI
2ON USED FOR ALL MASSES THAT ELEMENT - BLANKS DENOTE OP
XTION NOT EFF
3ECTED'//)
106 FORMAT(' MASS=',I4,A1,T11,'AT.NO.=',I3,I22,'NAME=',A2)
107 FORMAT('+',T32,'MU(1/SEC)='E14.8)
108 FORMAT('+',T58,'X-SECTION(BARNS)='E14.8)
109 FORMAT('+',T91,'LAMBDA(1/SEC)='E14.8)
110 FORMAT('1 END SUBROUTINE CROSS - CALCULATION OF MU')
END

```

```
/*
```

```

SUBROUTINE ZEROF5
COMMON PTR4,PTR5
DIMENSION A(11),B(11),Q(45)
INTEGER PTR5
EQUIVALENCE (A(1),Q(6)),(B(1),Q(21))
DO 10 I=1,42
DO 10 J=1,25
PTR5=(I-1)*25+J
READ(5,PTR5) Q
PTR5=PTR5-1
FIND (5,PTR5)
DO 5 K=1,11
A(K)=0.
B(K)=0.
5 CONTINUE
WRITE(5,PTR5) Q
10 CONTINUE
RETURN
END

```

```
/*
```

```

SUBROUTINE CALC
COMMON PTR4,PTR5,X9(11),COOLTM(11),X8(24),COULMX,X7(42
X),BURNTM
INTEGER PTR4,PTR5,BRNCH,DCHK,ONE/1/,COULMX
REAL LMDA(15),LMDAFL(2),MU(15), MSTATE
DIMENSION MNCHK(45),R(15),YLD(15),BEF(15),ANS(2,11),ST
XATE(2)
DIMENSION AFF(11),MAXN(15),Q(17)
DATA MSTATE,ASTATE,BSTATE/'M','A','B'/
NATOML=26
PTR4=1
PTR5=1

```

```
C
```

```
C ZERO OUT MNCHK FILE
```

```
DO 20 I=1,45
```

```
20 MNCHK(I)=0
```

```
C
```

```

C  READ AFIL  FOR CHAIN, MAX BRANCHES, NODES EACH BRANCH
  1 READ(4*PTR4) MASSC,MAXB,MAXN
    IF(MASSC.EQ.999) GO TO 1000
C
C  BEGIN BRANCH AND NODE DO LOOPS
  DO 902 KKK=1,MAXB
    KK=MAXN(KKK)
    DO 900 I=1,KK
C
C  READ NODE DATA
  READ(4*PTR4) MASS, BRNCH, NODE, LMDA(I), MU(I), R(I), YLD(I),
    XNAME, STATE1,
    INATOM, (Q(J), J=1, 7)
C
C  CHECK FOR 51 SB 124M
  IF(MASS.NE.124) GO TO 30
  IF(INATOM.NE.51) GO TO 30
  IF(STATE1.NE.MSTATE) GO TO 30
  PTR5=1050
  GO TO 35
C
C  CALCULATE RELATIVE RECORD NO.
  30 NBRR=INATOM-26
    IF(MNCHK(NBRR).EQ.0) MNCHK(NBRR)=MASS
    MASSR=MASS-MNCHK(NBRR)+1
    PTR5=25*(NBRR-1)+MASSR
  35 FIND(5*PTR5)
C
C  BEGIN ACTUAL BURNUP CALCULATIONS
  150 BAN=0.
C
C  SIGMA SUMMATION  M=1 TO I
  DO 475 M=1,I
    II=I
    SUM=0.
C
C  CHECK  KK AND/OR M =I  IF SO DIFFERENT EQUATIONS
  IF(I.NE.KK) GO TO 398
    II=II-1
  IF(I.NE.M) GO TO 398
    BAN=100.*R(M)*YLD(M)+BAN
  GO TO 475
C
C  SIGMA SUM  J=1 TO I OR I-1
  398 DO 450 J=M,II
    DPOD=1.
C
C  PI SUM UNDER SIGMA SUM ON J
  DO 401 K=M,I
    IF(K.EQ.J) GO TO 401
    HLDDD=MU(K)-MU(J)
    IF(HLDDD.EQ.0.) HLDDD=6.3

```

```

      DPCD=DPOD*HLDD
401 CONTINUE
C
C   END   PI SUM
C   CHECK SATURATION FACTORS TO ASSUMPTIONS
      EXPP=MU(J)*BUPNTM
      IF(EXPP.LT.1.E-04) GO TO 405
      IF(EXPP.GT.10.) GO TO 404
      EXPP=1.-EXP(-EXPP)
      GO TO 405
404 EXPP=1
405 SUM=SUM+EXPP/( MU(J)*DPOD)
450 CONTINUE
C
C   END OF SIGMA SUM ON J
C   PI SUM OF LAMBDA S
      L=I-1
      PIAMD=1
      IF(M.GT.L) GO TO 471
      DO 470 LL=M,L
470 PIAMD=PIAMD*LMDA(LL)
C
C   FINAL SUMS
471 IF(I.EQ.KK) GO TO 474
      BAN=BAN+YLD(M)*R(M)*(PIAMD*SUM)*100./BURNTM
      GO TO 475
474 BAN=BAN+YLD(M)*(((100./BURNTM)*R(M)*(PIAMD*SUM))+(100.
      X*R(M)))
475 CONTINUE
C
C   END OF BURNUP TIME CALCULATIONS
C   CHECK AND TEMP STORE ANSWERS
      IF(BAN.LE.0.) GO TO 900
      BFF(I)=BAN
C
C   END   BURNUP   CSECT - BEGIN   COOL TIME CSECT
C
      DO 601 LC=1,COOLMX
      IF(COOLTM(LC).NE.0.) GO TO 502
      AFF(LC)=BAN
      GO TO 806
502 SSUM=0.
C
C   SIGMA SUM   M=1 TO I
      DO 600 M=1,I
      ASUM=0.
C
C   SIGMA SUM   J=M TO I
      DO 550 J=M,I
C
C   PI SUM OVER K=1 TO I AND CHECK ATTENUATION FACTOR
      DPOD=1

```

```

      DO 525 K=M,I
      IF(K.EQ.J) GO TO 525
      HLDDD=LMDA(K)-LMDA(J)
      IF(HLDDD.EQ.0.) HLDDD=6.3
      DPOD=DPOD*HLDDD
525  CONTINUE
      EXPP=EXP(-LMDA(J)*COOLTM(LC))
C
C  END PI SUM
550  ASUM=ASUM+EXPP/DPOD
C  END SIGMA SUM OVER J
C  PI SUM OF LAMBDAS K=1 TO I-1 AND CHECK VALUE
      L=I-1
      PIAMD=1.
      IF(M.GT.L) GO TO 600
      DO 575 LL=M,L
575  PIAMD=PIAMD*LMDA(LL)
C  END PI SUM
600  SSUM=SSUM+BFF(M)*(ASUM*PIAMD)
      IF(SSUM.LT.0.) SSUM=0.
601  AFF(LC)=SSUM
      DO 620 LC=1,COOLMX
      IF(AFF(LC).GT.0.) GO TO 621
620  CONTINUE
      GO TO 900
621  CONTINUE
C  END COOLING TIME CSECT
C
C  WRITE ANSWERS TO DISK
806  READ(5,PTR5)NATOM,NAME,MASS,STATE(1),LMDAFL(1),(ANS(1,
      XL),L=1,11),
      IDCHK,NREC,STATE(2),LMDAFL(2),(ANS(2,L),L=1,11),(Q(J),J=
      X=1,14)
      PTR5=PTR5-1
      FIND(5,PTR5)
C
C  CHECK STATE OF ELEMENT K= POSITION IN RECORD L=SUBRECORD
      XS WRITTEN
      K=1
      IF((STATE1.EQ.MSTATE).OR.(STATE1.EQ.BSTATE)) K=2
      DO 650 LC=1,COOLMX
650  ANS(K,LC)=ANS(K,LC)+AFF(LC)
      STATE(K)=STATE1
      LMDAFL(K)=LMDA(I)
      WRITE(5,PTR5)NATOM,NAME,MASS,STATE(1),LMDAFL(1),(ANS(1,
      X,L),L=1,11),
      IONE,NREC,STATE(2),LMDAFL(2),(ANS(2,L),L=1,11),(Q(J),J=
      X1,14)
900  CONTINUE
902  CONTINUE
      GO TO 1
1000 PTR5=1

```

```

PTR4=1
WRITE(3,800)
RETURN
800 FORMAT(// ' NOTE:COMPLETION - CALC' )
END

/*
SUBROUTINE OUTPUT
COMMON PTR4,PTR5,X9(22),CLTM(22),BURNTM(2),LCMAX,X8(4)
X,
IPGTITL(8),IPOUT
DIMENSION SUM(11),STATE(2),ANS(2,11),Q(14),LLL(2),HLD
XMAS(161,11),
INXASS(161)
REAL LMDA(2)
INTEGER DCHK,PTR5
DATA SEC,AMIN,HRS,DAYS,YRS,ALL/'S','M','H','D','Y','AL
XL'//,
AHLDMAS/1771*0./,NMASS/161*0./,IPLACE,IPAGE/5,1/,KPTR/0/
C   LCMAX = MAX COOLING TIMES
C   BURNTM= ALPHA BURN TIME CONSIDERED
C   PGTITL= ALPHA TYPE REACTION
C   CLTM= ALPHA ALL COOLING TIMES
C   IPOUT = TYPE OF OUTPUT DESIRED
C   1-ALL ELEMENTS 2-BY MASS NO. 3-BY ATOMIC NO.
C   ICASE = SPECIAL OPTION 1-TAKEN 0-NOT TAKEN
C   ICASEN= NO. SPECIAL OPTION OUTPUT
PTR5=1
ADY=3600.*24.
AYR=ADY*365.
500 WRITE(3,200) IPAGE
WRITE(3,201) PGTITL,BURNTM
GO TO (501,502,503),IPOUT
501 WRITE(3,202) ((CLTM(2*K-1),CLTM(2*K)),K=1,LCMAX)
GO TO 504
502 WRITE(3,210) ((CLTM(2*K-1),CLTM(2*K)),K=1,LCMAX)
GO TO 504
503 WRITE(3,211) ((CLTM(2*K-1),CLTM(2*K)),K=1,LCMAX)
504 LINE=5
IPAGE=IPAGE+1
GO TO (48,49,90,404,5),IPLACE
5 DO 400 KKK=27,66
DO 10 I=1,11
10 SUM(I)=0.
KCHK=0
NCHK=1
DO 390 KK=1,25
12 READ(5,PTR5) NATOM,NAME,MASS,STATE(1),LMDA(1),(ANS(1,J
X),J=1,11),
1DCHK,NREC,STATE(2),LMDA(2),(ANS(2,L),L=1,11),Q
DO 90 I=1,2
IF(DCHK.NE.1) GO TO 48
LLL(I)=0

```



```

      DO 15 J=1,LCMAX
      IF(ANS(I,J).GT.0.) LLL(I)=1
15  CONTINUE
      IF(LLL(I).EQ.0) GO TO 48
      DO 20 J=1,LCMAX
20  SUM(J)=SUM(J)+ANS(I,J)
      GO TO (30,50,70),IPOUT
30  IF(LMDA(I).EQ.0.) GO TO 45
35  THALF=.693/LMDA(I)
      IF(THALF.GT.60.) GO TO 36
      TM=SEC
      GO TO 40
36  IF(THALF.GT.3600.) GO TO 37
      TM=AMIN
      THALF=THALF/60.
      GO TO 40
37  IF(THALF.GT.ADY) GO TO 38
      TM=HRS
      THALF=THALF/3600.
      GO TO 40
38  IF(THALF.GT.AYR) GO TO 39
      TM=DAYS
      THALF=THALF/ADY
      GO TO 40
39  THALF=THALF/AYR
      TM=YRS
40  IF(LINE.EQ.5.OR.NCHK.EQ.1) GO TO 43
      WRITE(3,203) MASS,STATE(I),THALF,TM,(ANS(I,J),J=1,LCMA
      XX)
      GO TO 47
43  WRITE(3,204) NATOM,NAME,MASS,STATE(I),THALF,TM,
      I(ANS(I,J),J=1,LCMAX)
      GO TO 47
45  IF(LINE.EQ.5.OR.NCHK.EQ.1) GO TO 46
      WRITE(3,205) MASS,STATE(I),(ANS(I,J),J=1,LCMAX)
      GO TO 47
46  WRITE(3,206) NATOM,NAME,MASS,STATE(I),(ANS(I,J),J=1,LC
      XMAX)
47  NCHK=NCHK+1
      LINE=LINE+1
      IPLACE=1
      IF(LINE.GE.58) GO TO 500
48  IF(KK.LT.25) GO TO 49
      KCHK=KCHK+1
      IF(KCHK.NE.2) GO TO 49
      LINE=LINE+2
      WRITE(3,207) NATOM,NAME,(SUM(J),J=1,LCMAX)
      IPLACE=2
      WRITE(3,213)
      IF(LINE.GE.58) GO TO 500
49  GO TO 90
50  DO 53 J=1,LCMAX

```

```

53 HLDMAS(MASS,J)=HLDMAS(MASS,J)+ANS(I,J)
   NMASS(MASS)=NMASS(MASS)+1
   GO TO 90
70 NCHK=NCHK+1
   IF(KK.LT.25) GO TO 90
   WRITE(3,208) NATOM,NAME,ALL,NCHK,(SUM(J),J=1,LCMAX)
   IPLACE=3
   LINE=LINE+2
   IF(LINE.GE.58) GO TO 500
90 CONTINUE
   IF(NATOM.NE.51) GO TO 390
   IF(MASS.NE.124) GO TO 390
   IF(KPTR.EQ.1) GO TO 360
   KPTR=1
   KHLD=PTR5
   PTR5=1050
   GO TO 12
360 PTR5=KHLD
   KPTR=0
390 CONTINUE
400 CONTINUE
   IF(IPCUT.NE.2) GO TO 450
   IPLACE=4
   DO 405 I=72,161
   WRITE(3,209) ALL,I,NMASS(I),(HLDMAS(I,J),J=1,LCMAX)
   LINE=LINE+2
   IF(LINE.GE.58) GO TO 500
404 CONTINUE
405 CONTINUE
450 WRITE(3,212)
   RETURN
200 FORMAT(T1,'1',T64,'TABLE',I3)
201 FORMAT(T24,'FISSION PRODUCT SPECTRA FOR CA. ',A4,' ('
      X,A4,' ',
      1A4,' IRRIDIATION)'/T43,'ATOMS PER 200 ATOMS FISSIONED
      XAFTER COOLIN
      2G FOR'/)
202 FORMAT(T3,'ELE',T8,'MASS',T14,'HALF-LIFE',T24,11(' ',A
      X4,' ',A4))
203 FORMAT(T7,I4,A1,T13,E9.3,A1,T13,' ',T24,1P11E10.3)
204 FORMAT(T1,I3,T5,A2,T7,I4,A1,T13,E9.3,A1,T13,' ',T24,1P
      X11E10.3)
205 FORMAT(T7,I4,A1,T15,'STABLE',T24,1P11E10.3)
206 FORMAT(T1,I3,T5,A2,T7,I4,A1,T15,'STABLE',T24,1P11E10.3
      X)
207 FORMAT(T4,'TOTALS FOR',I3,1X,A2,' =',T24,1P11E10.3)
208 FORMAT(T1,I3,T5,A2,T8,A3,T16,I3,T24,1P11E10.3)
209 FORMAT(T3,A3,T8,I4,T16,I3,T24,1P11E10.3)
210 FORMAT(T3,'ELE',T8,'MASS',T14,'ISOTOPES',T24,11(' ',A4
      X,' ',A4))
211 FORMAT(T3,'ELE',T8,'MASS',T13,'AT. NO. 'S',T24,11(' ',
      XA4,' ',A4))

```

212 FORMAT('1',T14,'END SUBROUTINE OUTPUT')

213 FORMAT(' ')

END

/*

SUBROUTINE SPCASE(ICASEN)

COMMON PTR4,PTR5,X9(22),CLTM(22),BURNTM(2),LCMAX

DATA BLK/' ','S/'/'S'/

DIMENSION STATE(2),ANS(2,11),Q(14)

INTEGER PTR5,DCHK

REAL LMDA(2)

DATA IPLACE,IPAGE/1,1/

C VARIABLES AS IN CALC AND OUTPUT

C VAR ARE BURNTM,LCMAX,CLTM, ALL OF FILE 5

20 WRITE(3,100) IPAGE,BURNTM

WRITE(3,101) ((CLTM(2*K-1),CLTM(2*K)),K=1,LCMAX)

LINE=5

IPAGE=IPAGE+1

GO TO (21,50),IPLACE

21 IPLACE=2

DO 99 I=1,ICASEN

READ(1,107) NATOM,MASS

PTR5=(NATOM-27)*25+1

FIND (5*PTR5)

LL=0

IF(NATOM.NE.51) GO TO 25

LL=LL+1

IF(MASS.NE.124) GO TO 25

LL=LL+1

25 READ(5*PTR5) NATOM1,NAME,MASS1

MASSR=MASS-MASS1

IF(MASSR.GE.0) GO TO 27

WRITE(3,105) NATOM,NAME,MASS

GO TO 99

27 PTR5=PTR5-1+MASSR

30 READ(5*PTR5) NATOM,NAME,MASS,STATE(1),LMDA(1),(ANS(1,J)

X),J=1,11),

ADCHK,NREC,STATE(2),LMDA(2),(ANS(2,K),K=1,11)

DO 51 J=1,2

IF(DCHK.NE.1) GO TO 62

DO 35 K=1,LCMAX

IF(ANS(J,K).NE.0.) GO TO 36

35 CONTINUE

GO TO 48

36 WRITE(3,102) NATOM,NAME,MASS,STATE(J),LMDA(J),S,

1(ANS(J,K),K=1,LCMAX)

GO TO 49

48 IF(LL.EQ.3) GO TO 51

IF(STATE(J).EQ.BLK.AND.J.EQ.2) GO TO 49

WRITE(3,103) NATOM,NAME,MASS,STATE(J)

49 LINE=LINE+1

IF(LINE.GE.58) GO TO 20

50 IF(LL.EQ.3) GO TO 51

```

      IF(LL.EQ.2) GO TO 60
51  CONTINUE
99  CONTINUE
      WRITE(3,104)
      RETURN
60  PTR5=1050
      LL=3
      GO TO 30
62  WRITE(3,106) NATOM,NAME,MASS
      GO TO 99
100 FORMAT(T1,'1',T64,'TABLE',I3/T54,'ISOTOPES OF SPECIAL
      XINTEREST'/
      CT54,'(IRRADIATION TIME=',A4,' ',A4,')'/)
101 FORMAT(T3,'ELE',T8,'MASS',T14,'DCY CONST',T24,11(' ',A
      X4,' ',A4))
102 FORMAT(T1,I3,T5,A2,T7,I4,A1,T13,1PE9.2,A2,T24,11F10.3)
103 FORMAT(T1,I3,T5,A2,T7,I4,A1,T48,'THIS STATE HAS CONCEN
      XTRATIONS OF
      XZERO FOR ALL COOLING TIMES')
104 FORMAT(T1,'1',T14,'END SUBROUTINE SPECIAL ISOTOPE OUTP
      XUT')
105 FORMAT(T1,I3,T5,A2,T7,I4,T44,'THIS ELEMENT NOT FOUND I
      XN ANY DECAY
      XSCHEME - ELEMENT OUTPUT IGNORED')
106 FORMAT(T1,I3,T5,A2,T7,I4,T44,'THIS ENTIRE ELEMENT HAS
      XCONCENTRATIO
      XNS OF ZERO FOR ALL COOLING TIMES')
107 FORMAT(T12,I2,T24,I3)
      END
/*
      SUBROUTINE HEAT
      COMMON PTR4,PTR5,X9(22),CLTM(22),BURNTM(2),X8(14),IHEA
      XT,ICOLM,
      IIC(11),NBCDS
      INTEGER DCHK,PTR4,PTR5
      DIMENSION A(11),BANS(11)
      DATA BANS/11*0./,WCONV/1./
      WRITE(3,101) BURNTM
      IF(IHEAT.EQ.1) GO TO 200
      IPLACE=1
      WRITE(3,102)
      WRITE(3,104) ((CLTM(2*IC(K)-1),CLTM(2*IC(K))),K=1,IC0
      XOLM)
      DO 30 I=1,NBCDS
5  READ(1,100) NATM,MAS
      PTR5=(NATM -27)*25+1
      READ(5,PTR5) NATOM,NAME,MASS
      MASSR=MAS -MASS
      IF(MASSR.GE.0) GO TO 8
      WRITE(3,107) NATOM,NAME,MAS
      GO TO 30
8  PTR5=PTR5-1+MASSR

```

```

      CALL      HEAT2(A,BANS,PTR5,IC,ICOOCLM,WCONV,NATOM,MAS
XS)
20  WRITE(3,106)  NATOM,NAME,MASS,(A(K),K=1,ICOOCLM)
30  CONTINUE
    GO TO 500
200  IPLACE=2
    WRITE(3,103)
    WRITE(3,104)  ((CLTM(2*IC(K)-1),CLTM(2*IC(K))),K=1,IC
XOOCLM)
    DO 205  I=1,1050
      PTR5=I
      CALL      HEAT2(A,BANS,PTR5,IC,ICOOCLM,WCONV,NATOM,MAS
XS)
204  CONTINUE
205  CONTINUE
500  WRITE(3,105)  (BANS(K),K=1,ICOOCLM)
      PTR5=1
      PTR4=1
      WRITE(3,108)
      RETURN
100  FORMAT(T12,I2,T20,I3)
101  FORMAT('1',T41,'BETA HEATING OPTION EFFECTED (' ,A4,' '
X,A4,
1' IRRIDIATION)'/T51,'OUTPUT IS PER 200 ATOMS FISSIONED
X')
102  FORMAT(/T41,'SELECTED ELEMENTS LISTED FOR BETA DECAY I
XN MEV
1//T3,'ELE',T8,'MASS')
103  FORMAT(/T47,'TOTAL HEATING FOR BETA DECAY IN MEV
X'///)
104  FORMAT('+',T13,'COOL TIMES=',T24,11(' ',A4,' ',A4))
105  FORMAT(T4,'TOTAL BETA HEATING=',T24,1P11E10.3)
106  FORMAT(T1,I3,T5,A2,T7,I4,T24,1P11E10.3)
107  FORMAT(T1,I3,T5,A2,T7,I4,T27,'ELEMENT NOT FOUND IN ANY
X DECAY SCHEM
1E-ELEMENT DELETED')
108  FORMAT('1')
      END

```

/*

```

      SUBROUTINE HEAT2(A,BANS,PTR5,IC,ICOOCLM,WCONV,NATOM,MAS
XS)
      DIMENSION A(11),BANS(11),IC(11)
      INTEGER PTR5,DCHK
      REAL LMDA(2)
      DIMENSION Q(14),CT(2,11),NDCY(2),RD(2),B1(2),B2(2)
      EQUIVALENCE (Q(1),NDCY(1)),(Q(3),RD(1)),(Q(5),B1(1)),
X(Q(9),B2(1))
300  READ(5,PTR5) NATOM,NAME,MASS,ST,LMDA(1),(CT(1,L),L=1,1
X1),DCHK,NRR,
1SI,LMDA(2),(CT(2,L),L=1,11),Q
    DO 306  LC=1,ICOOCLM
306  A(LC)=0.

```

```

      IF(DCHK.NE.1) GO TO 334
      IF(NDCY(1).EQ.0) GO TO 321
      DO 320 LC=1,ICQOLM
      HLD=CT(1,IC(LC))-CT(1,IC(LC))*EXP(-LMDA(1))
      A(LC)=HLD*RD(1)*R1(1)*WCONV+A(LC)
      IF(NDCY(1).EQ.1) GO TO 320
      A(LC)=A(LC)+HLD*(1.-RD(1))*R1(2)*WCONV
320  CONTINUE
321  IF(NDCY(2).EQ.0) GO TO 331
      DO 330 LC=1,ICQOLM
      HLD=CT(2,IC(LC))-CT(2,IC(LC))*EXP(-LMDA(2))
      A(LC)=A(LC)+HLD*RD(2)*R2(1)*WCONV
      IF(NDCY(2).EQ.1) GO TO 330
      A(LC)=A(LC)+HLD*(1.-RD(2))*R2(2)*WCONV
330  CONTINUE
331  DO 333 LC=1,ICQOLM
333  BANS(LC)=BANS(LC)+A(LC)
334  RETURN
      END

```

/*

```

      SUBROUTINE SHIELD
      COMMON PTR4,PTR5,X9(22),CLTM(22),BURNM(2),X9(28),ISHL
      XD,
      LIS(11),ISMAX,IGPMAX
      INTEGER DCHK,PTR4,PTR5
      REAL LMDA(2)
      DIMENSION NDCY(2),RD(2),G(2,2),CT(2,11),GPMAX(80),
      IGPMIN(80),IGP(80),ANS(80,11),ZRD(2),ZR(2)
      DATA ANS/880*0./
      DO 10 I=1,IGPMAX
10  READ(1,100) IGP(I),GPMAX(I),GPMIN(I)
      DO 200 I=1,1050
      PTR5=I
      READ(5*PTR5) NATOM,NAME,MASS,ST,LMDA(1),(CT(1,L),L=1,1
      X1),DCHK,NRR,
      ASI,LMDA(2),(CT(2,L),L=1,11),NDCY,RD,ZRD,(G(1,L),L=1,2)
      X,ZR,
      B(G(2,L),L=1,2),QQ,QQQ
      IF(DCHK.NE.1) GO TO 200
      DO 90 K=1,2
      IF(NDCY(K).EQ.0) GO TO 90
      DO 20 J=1,IGPMAX
      IF(G(K,1).GE.GPMIN(J).AND.G(K,1).LE.GPMAX(J)) GO TO 18
      GO TO 20
18  DO 19 LC=1,ISMAX
19  ANS(IGP(J),LC)=ANS(IGP(J),LC)+RD(K)*(CT(K,IS(LC))-CT(K
      X,IS(LC))*
      IEXP(-LMDA(K)))
      GO TO 21
20  CONTINUE
21  IF(NDCY(K).EQ.1) GO TO 90
      DO 40 J=1,IGPMAX

```

```

      IF (G(K,2).GE.GPMIN(J).AND.G(K,2).LE.GPMAX(J)) GO TO 38
      GO TO 40
38 DO 39 LC=1,ISMAX
39 ANS(IGP(J),LC)=ANS(IGP(J),LC)+(1.-PD(K))*(CT(K,IS(LC))
X-
      ICF(K,IS(LC))*EXP(-LMDA(K)))
40 CONTINUE
90 CONTINUE
200 CONTINUE
      WRITE(3,101) BURNTM
      WRITE(3,102) ((CLTM(2*IS(I)-1),CLTM(IS(I)*2)),I=1,ISMA
XX)
      DO 220 J=1,IGPMAX
      DO 218 I=1,IGPMAX
      IF(J.NE.IGP(I)) GO TO 218
      WRITE(3,103) I, GPMIN(I), GPMAX(I),(ANS(I,IS(LC)),LC=1
X,ISMAX)
      GO TO 220
218 CONTINUE
220 CONTINUE
      RETURN
100 FORMAT(T11,I2,T26,F6.2,T48,F6.2)
101 FORMAT('1',T40,'GAMMA ENERGY GROUP OPTION EFFECTED(',A
X4,' ',A4,
1' IRRADIATION)'/T41,'OUTPUT IS ATOMS PER 200 ATOMS FIS
XSIGNED IN EN
ERGY GROUP')
102 FORMAT(/T8,'LOWER',T17,'UPPER'/T8,'ENERGY',T17,'ENERGY
X'/T2,'GP.',
AT3,'BOUND',T17,'BOUND',T26,'COOLING TIMES = '/T2,'NO.
X',T8,'(MEV)'
B,T17,'(MEV)',T24,11(' ',A4,' ',A4))
103 FORMAT(T1,I3,T6,F7.3,T15,F7.3,T24,1P11E10.3)
104 FORMAT('1',T12,'END SUBROUTINE SHIELD')
      END

```

/*

```

C THIS IS PROGRAM ENDZAP
  DEFINE FILE 4(2400,17,U,I4)
  DEFINE FILE 5(1500,45,U,I5)
  DIMENSION NNN(15),III(3),BG(4),N5(6),Q(45),XS(42,109,2
X),YLD(180)
  I4=1
20 READ(4,I4) MSS,MAXB,NNN
  WRITE(8      ) MSS,MAXB,NNN
  IF(MSS.EQ.999) GO TO 24
  M=0
  DO 21 I=1,MAXB
21 M=M+NNN(I)
  DO 22 I=1,M
    READ(4,I4)      III,BG,N1,N2,N3,N4,N5
    WRITE(8      ) III,BG,N1,N2,N3,N4,N5
22 CONTINUE
  GO TO 20
24 END FILE 8
  I5=1
  DO 25 I=1,1050
    READ(5,I5) Q
    WRITE(8      ) Q
25 CONTINUE
  END FILE 8
  DO 44 L=1,2
  DO 44 I=1,42
    READ(12)      (XS(I,J,L),J=1,109)
    WRITE(8      ) (XS(I,J,L),J=1,109)
44 CONTINUE
  END FILE 8
  DO 51 KK=7,10
    KKK=KK+7
    READ(KKK) YLD
    WRITE(8      ) YLD
  END FILE 8
51 CONTINUE
  DO 55 I=1,2
    READ(18) YLD
    WRITE(8      ) YLD
55 CONTINUE
  READ(18) YLD
  WRITE(8)      YLD
  END FILE 8
  WRITE(3,150)
  CALL EXIT
103 FORMAT(17I4)
104 FORMAT(3I4,4E10.4,2A4,2I4,6I2)
107 FORMAT(I4,A4,I4,A4,12E10.4,2I4,A4,12E10.4,2I4,12E10.4)
109 FORMAT(40(5E12.4))
112 FORMAT(60(30I4))
150 FORMAT('1',T10,'END OF ENDZAP')
  END

```

/*

C THIS IS CHKZAP

```

    DIMENSION PGTITL(8),CLTM(22),COOLTM(11),BURNTM(11),BTM
    X(22),IC(11),
    LIS(11)
    REAL C(80)
    WRITE(3,103)
30  READ(9,100,END=99) C
    WRITE(3,101) C
    GO TO 30
99  READ(11) IER,IYLD,I PROG,I OBL,IMU,PGTITL,CLTM,COOLTM,LC
    XMAX,BURNTM,
    2BTM,LBMAX,IPOUT,ICASE,ICASEN,IHEAT,ICOOLM,IC,NBCDS,ISH
    XLD,IS,
    3ISMAX,IGPMAX
    WRITE(3,104)
    WRITE(3,102) IER,IYLD,I PROG,I OBL,IMU,PGTITL,CLTM,COOLT
    XM,LCMAX,
    4BURNTM,BTM,LBMAX,IPOUT,ICASE,ICASEN,IHEAT,ICOOLM,IC,NB
    XCDS,ISHLD,
    5IS,ISMAX,IGPMAX
    CALL EXIT
100 FORMAT(80A1)
101 FORMAT(' ',80A1)
102 FORMAT(5(2X,I4),8A4/2X,11(2A4,2X)/2X,11(F10.4,1X)/2X,I
    X4/11(F10.4,1
    6X)/2X,11(2A4,2X)/19(2X,I4)/13(2X,I4))
103 FORMAT('!OUTPUT FROM 9 (80A1)')
104 FORMAT('!OUTPUT FROM 11')
    END

```

/*

C THIS IS F4BUG

```

    DIMENSION N(17)
    INTEGER PTR4
    DEFINE FILE 4(2400,17,U,PTR4)
    PTR4=15

```

C FIRST WORD IN RECORD 15 SHOULD BE 77 - IS CHANGED HERE TO
X 999

```

    READ(4,PTR4) N
    N(1)=77
    PTR4=15
    WRITE(4,PTR4) N
    CALL EXIT
    END

```

/*

C THIS IS PFILE5

```

    DIMENSION Q(45)
    INTEGER PTR5
    DEFINE FILE 5(1500,45,U,PTR5)
    DO 50 I=5,8
    DO 10 J=1,5
    PTR5=(I-1)*25+J

```

```

      READ(5,PTR5) Q
      WRITE(3,101) (Q(L),L=1,31)
      WRITE(3,100) (Q(L),L=1,3),(Q(L),L=32,43)
10  CONTINUE
50  CONTINUE
      CALL EXIT
100 FORMAT(I3,A2,I4,2I2,10F10.3)
101 FORMAT(I3,A2,I4,A1,12E10.4/2I2,A1,12E10.4)
      END

```

```

/*

```

```

C  THIS IS DCYPRP
      DIMENSION Z(90),GAM(90),DEL(90),L(4),K(4),C(4),D(4)
      DIMENSION OUT(31)
      INTEGER PTR5
      DATA BLK/'      '/,L/122,128,146,152/,K/51,53,61,63/
      DEFINE FILE 5(1500,45,U,PTR5)
      DO 10 I=72,161
      B=I
      GAM(I-71)=(1.+B**(.6667)/128.)*76.*2./9
      Z(I-71)=77./GAM(I-71)
10  DEL(I-71)=(33.5/(B**(.75)))*2.
      DO 5 I=1,4
      B=L(I)
      Q=GAM(L(I)-71)*(B-Z(L(I)-71)-.5)
      Q=ABS(Q+DEL(L(I)-71))
      B=Z(L(I)-71)
      IF(ABS(B-K(I)).LE.1) GO TO 2
      IF(ABS(B-K(I)).LE.2) GO TO 3
      C(I)=Q
      D(I)=0
      GO TO 5
2  C(I)=.85*Q
   D(I)=.15*Q
   GO TO 5
3  C(I)=.45*Q
   D(I)=.55*Q
5  CONTINUE
   M=72
12  READ(1,100,END=999) M1,TI,N,NR1,NR2,R1,R2
      IF(M1.NE.0) M=M1
      KKK=0
      IF(M.NE.99) GO TO 14
      M=51
      KKK=1
14  B=M
      PTR5=(M-27)*25+1
      READ(5,PTR5) NATOM,NAME,MASS
      MASSR=N-MASS
      PTR5=PTR5-1+MASSR
      FIND (5,PTR5)
      Q=GAM(N-71)*(Z(N-71)-B-.5)
15  IF(MOD(2,N).EQ.1) GO TO 20

```

```

      IF(MOD(2,M).EQ.1) Q=Q+2.*DEL(N-71)
      Q=Q-DEL(N-71)
20  Q=ABS(Q)
      B=M
      IF(ABS(B-Z(N-71)).LE.1) GO TO 25
      IF(ABS(B-Z(N-71)).LE.2) GO TO 27
      R=0.5
      GO TO 30
25  Q=.85*Q
      R=.15*Q
      GO TO 30
27  Q=.45*Q
      R=.55*Q
30  B11=0.
      READ(5'PTR5) OUT
      PTR5=PTR5-1
      FIND(5'PTR5)
      B12=0.
      B21=0.
      B22=0.
      G11=0.
      G12=0.
      G21=0.
      G22=0.
      IF(M.EQ.51.AND.N.EQ.122) GO TO 35
      IF(M.EQ.53.AND.N.EQ.128) GO TO 36
      IF(M.EQ.61.AND.N.EQ.146) GO TO 37
      IF(M.EQ.63.AND.N.EQ.152) GO TO 38
      GO TO 41
35  B11=Q
      B12=C(1)
      G11=R
      G12=D(1)
      GO TO 66
36  B11=Q
      B12=C(2)
      G11=R
      G12=D(2)
      GO TO 66
37  B11=Q
      B12=C(3)
      G11=R
      G12=D(3)
      GO TO 66
38  B11=Q
      B12=C(4)
      G11=R
      G12=D(4)
      B21=Q
      B22=C(4)
      G21=R
      G22=D(4)

```

```

      GO TO 66
41  IF(NR1.EQ.0) GO TO 55
      B11=0
      G11=R
      IF(NR1.EQ.1) GO TO 55
      B12=ABS(Q-.31)
      G12=R
55  IF(NR2.EQ.0) GO TO 65
      IF(T1.EQ.BLK) GO TO 58
      B21=ABS(Q-.25)
      G21=R
57  IF(NR2.EQ.1) GO TO 65
      B22=ABS(Q-.25)
      G22=R
      GO TO 65
58  G21=.5*Q
      IF(Q.GE.1.) G21=1./Q
      GO TO 57
65  IF(KKK.EQ.1) M=99
66  WRITE(3,101) M,N,NR1,NR2,R1,R2,B11,B12,G11,G12,B21,B22
      X,G21,G22
      WRITE(5,PTR5)OUT,NR1,NR2,R1,R2,B11,B12,G11,G12,B21,B22
      X,G21,G22
      GO TO 12
999 CALL EXIT
100 FORMAT(I2,A1,I3,2I2,2F5.2)
101 FORMAT(2X,I2,1X,I3,2I2,10E10.3)
      END

```

```

/*

```

```

C   THIS IS DATA1
C   THIS ROUTINE INITIALIZES AND READYS FILES 4 AND 5 + Z
C   XHI,ZLOW,ZS

```

```

C

```

```

      DIMENSION NS(166,72),NODE(200),R(200),STATE(200),NATOM
X(200),
      INAME(72),MASLOW(72),ZERO(12),IZERO(2),ZS(180),MAXN(15)
X,0(16),
      2MNN(17),NAMEE(200),MASSA(200)
      REAL MU,LMDA(200)
      INTEGER BRNCH(200),ZHI(180),ZLOW(180),PTR4,PTR5
      DATA IZERO/2*0/,ZERO/12*0./,MASLOW/72*0/,BLK/' '/,NS/1
X1952*1/
      DATA NS(75,32) ,NS(77,32) ,NS(77,34) ,NS(78,32)
X ,
      ANS(79,34) ,NS(83,34) ,NS(83,36) ,NS(84,35) ,NS
X(85,36) ,
      BNS(83,37) ,NS(91,39) ,NS(93,41) ,NS(95,41) ,NS
X(97,41) ,
      CNS(99,43) ,NS(100,41) ,NS(102,43) ,NS(103,45)
X,
      DNS(104,45) ,NS(105,45) ,NS(106,45) ,NS(109,45)
X,

```

```

ENS(109,46) ,NS(109,47) ,NS(110,47) ,NS(111,47)
X ,
FNS(111,46) ,NS(113,47) ,NS(113,48) ,NS(115,47)
X ,
GNS(115,48) ,NS(115,49) ,NS(117,48) ,NS(117,49)
X ,
HNS(118,49) ,NS(119,48) ,NS(119,49) ,NS(120,49)
X ,
INS(121,49) ,NS(123,49) ,NS(123,50) ,NS(125,50)
X ,
JNS(125,52) ,NS(126,51) ,NS(127,50) ,NS(127,52)
X ,
KNS(128,51) ,NS(129,50) ,NS(129,52) ,NS(129,54)
X ,
LNS(130,51) ,NS(131,52) ,NS(131,54) ,NS(133,52)
X ,
MNS(133,54) ,NS(134,55) ,NS(135,54) ,NS(137,56)
X ,
NNS(148,62) ,NS(151,63)/60*2/ ,NS(152,63)/3/
DATA YLD,MU/2*0./,MEND,IS,MS/999,51,124/,MCHK/72/
DATA Q/16*0./

```

C DATA FOR NSTATE RECORD B NO.2

C

```

DATA NS(78,33),NS(107,47)/2*2/,PTR4,PTR5/2*1/
DATA NAME/72*' '/

```

C DATA FOR ZHI,ZLOW,ZSTABLE RECORD TYPE C

```

DATA ZS/69*0.,31.2,31.584,31.968,32.352,32.736,33.120
X,33.504,
A33.888,34.272,34.656,35.040,35.424,35.808,36.192,36.57
X6,36.960,
B37.344,38.164,38.555,38.945,39.336,40.188,40.585,40.98
X2,41.379,
C41.776,42.173,42.570,42.967,43.364,43.761,44.158,44.55
X5,44.952,
D45.349,45.746,46.143,46.540,46.937,47.334,47.731,48.12
X8,48.525,
E48.922,49.319,49.716,49.557,49.930,50.304,50.677,51.05
X1,50.750,
F51.100,51.450,51.800,52.150,52.500,52.850,53.200,53.55
X0,53.900,
G54.250,54.600,54.950,55.300,55.650,56.000,57.075,57.42
X9,57.782,
H58.136,59.228,59.585,55.942,60.299,60.656,61.013,61.37
X0,61.727,
I62.081,62.441,62.798,63.155,63.512,63.869,63.913,64.27
X7,64.640,
J65.001,65.080,65.450,65.820,66.820,66.560,66.930,67.30
X0,15*0./
DATA ZHI/71*0,3*32,33,3*34,35,34,35,3*36,37,3*38,39,3*
X40,41,5*42,
A4*44,45,3*46,47,46,47,46,2*48,49,48,50,48,4*50,51,52,5
X1,3*52,53,

```

```

B5*54,55,4*56,58,57,58,59,58,3*60,5*62,63,64,63,4*64,65
X,2*66,19*0/
DATA ZLOW/71*0,4*27,3*28,3*29,2*30,2*31,3*32,2*33,3*34
X,3*35,2*36,
C37,3*38,4*39,3*40,2*41,3*42,3*43,2*44,2*45,2*46,3*47,5
X*48,2*49,
D50,3*51,2*52,3*53,3*54,2*55,2*56,3*57,2*58,3*59,3*60,1
X9*0/

```

```

C
C WRITE ZHI, ZLOW, & ZSTABLE MASTER FILE
WRITE(18) ZLOW
WRITE(18) ZHI
WRITE(18) ZS
DEFINE FILE 4(2400,17,U,PTR4)
DEFINE FILE 5(1500,45,U,PTR5)
PTR5=1
PTR4=1
1 READ(1,600)MASSC,MAXB
IF(MASSC.EQ.MEND) GO TO 80
IF(MASSC.NE.MCHK) GO TO 400
DO 5 I=1,15
5 MAXN(I)=0
M=0
C READ CHAIN CARDS M GIVES TOTAL NODES THIS CHAIN
C
C READ BRANCH CARDS
DO 50 I=1,MAXB
READ(1,601)MASSB,MAXBC,MAXN(I)
IF(((MAXBC.NE.I).OR.(MASSB.NE.MASSC)) GO TO 401
K=MAXN(I)
C
C READ NODE CARDS AND STORE DATA IN CORE
DO 50 J=1,K
M=M+1
READ(1,602) MASSN,BRNCH(M),NODE(M),LNDA(M),P(M),
1NAMEE(M),MASSA(M),STATE(M),NATOM(M)
IF(((MASSN.NE.MASSB).OR.(BRNCH(M).NE.I).OR.(NODE(M).N
XE.J))GO TO 2
NNATOM=NATOM(M)
C
C HOLD DATA FOR RECORD TYPE A
NAME(NNATOM)=NAMEE(M)
IF(MASLOW(NNATOM).EQ.0) MASLOW(NNATOM)=MASSN
50 CONTINUE
C
C WRITE OUT RECORDS IN USEABLE FORM TO MASTER UNIT
WRITE(4,PTR4) MASSC,MAXB,MAXN
WRITE(3,700) MASSC,MAXB,MAXN
DO 60 I=1,M
MA=MASSA(I)
N=NATOM(I)
WRITE(3,701) MASSA(I),BRNCH(I),NODE(I),LNDA(I),"U,R(I)
X,YLD,

```

```

      INAMEE(I),STATE(I),N,NS(MA      ,N),(Q(L),L=1,6)
      WRITE(4*PTR4) MASSA(I),BRNCH(I),NODE(I),LMDA(I),YU,R(I
      X),YLD,
      INAMEE(I),STATE(I),N,NS(MA      ,N),(Q(L),L=1,6)
60  CONTINUE
      MCHK=MCHK+1
      GO TO 1
80  WRITE(4*PTR4) MEND,Q
C
C  WRITE OUT FORMAT FOR  A TYPE RECORDS
C
      DO 100 I=27,66
      M=MASLOW(I)-1
      WRITE(3,700) MASLOW(I)
      DO 100 J=1,25
      M=M+1
      WRITE(5*PTR5) I,NAME(I),M,BLK,ZERO,IZERO,BLK,ZERO,  IZ
      XERO,
      I(Q(L),L=1,12)
100  CONTINUE
      DO 105 I=1,49
      WRITE(5*PTR5) IZERO(1),BLK,IZERO(1),BLK,ZERO,IZERO,BLK,
      XZERO,  IZERO
      I(Q(L),L=1,12)
105  CONTINUE
      WRITE(5*PTR5) IS,NAME(IS),MS,BLK,ZERO,IZERO,BLK,ZERO,
      X IZERO,
      I(Q(L),L=1,12)
999  CALL EXIT
400  WRITE(3,603)MCHK
      GO TO 999
401  WRITE(3,604)MASSC,I
      GO TO 999
      2 WRITE(3,605)MASSC,I,J
      GO TO 999
600  FORMAT(I3,T10,I2)
601  FORMAT(I3,T10,I2,T21,I2)
602  FORMAT(I3,T10,I2,T21,I2,T48,E12.6,T62,F8.6,T71,A2,1X,I
      X3,A1,I3)
603  FORMAT(6X,'ERROR HALT - INPUT CHAIN',I4)
604  FORMAT(6X,'ERROR HALT - INPUT CHAIN',I4,'BRNCH',I2)
605  FORMAT(6X,'ERROR HALT-INPUT CHAIN',I4,'BRNCH',I2,'NODE
      X',I2)
700  FORMAT(17(2X,I4))
701  FORMAT(3(2X,I4),4(3X,E12.4),2(2X,A4),2(2X,I4),6F4.0)
      END
/*
C  THIS IS DATA1X
      DIMENSION YLD(180)
      DATA YLD/180*0./
      DO 60 K=14,17
      READ(1,99) (YLD(I),I=1,3)

```

```

      READ(1,100) (YLD(I),I=70,160)
      DO 15 I=70,160
15    YLD(I)=YLD(I)*1.E-02
      WRITE(K) YLD
60    CONTINUE
      RETURN
100  FORMAT(5E14.8)
99   FORMAT(3F9.2)
      END

```

```

/*

```

```

C   THIS IS PXSEC
      DIMENSION X(42,109,2)
      DO 10 L=1,2
      DO 10 I=1,42
10    READ(12) (X(I,J,L),J=1,109)
      DO 20 I=8,14
      DO 20 J=1,10
      JJ=J+71
      II=I+26
20    WRITE(3,100) II,JJ,(X(I ,J ,L),L=1,2)
      CALL EXIT
100  FORMAT(2(I4,2X),2(F16.4,3X))
      END

```

```

/*

```

```

C   THIS IS PRI4
      DIMENSION MAXN(15),Q(17)
      INTEGER PTR4
      DEFINE FILE 4(2400,17,U,PTR4)
      PTR4=1
      DO 50 I=1,89
      READ(4,PTR4) M,N,MAXN
      WRITE(3,700) M,N,MAXN
      MX=0
      DO 5 J=1,N
5     MX=MX+MAXN(J)
      DO 20 J=1,MX
      READ(4,PTR4) Q
      WRITE(3,701) Q
20    CONTINUE
50    CONTINUE
      CALL EXIT
700  FORMAT(17(2X,I4))
701  FORMAT(3(2X,I4),4(3X,E12.4),2(2X,A4),2(2X,I4),6F4.0)
      END

```

```

/*

```

```

C   THIS IS PYLD
      DIMENSION Y(180)
      DO 60 KK=14,17
      READ(KK) Y
      WRITE(3,100) (Y(I),I=1,3),(Y(J),J=72,79)
60    CONTINUE
      CALL EXIT

```



```

100 FORMAT(11E11.4)
END
/*
C   THIS IS CRSPRP
    DIMENSION X(42,109,2)
    INTEGER B
    DATA X/9156*0./,X(19,33,1)/40./,B/'B'/,E/'E'/
    DATA X(42,109,1)/15./
1   READ(1,100,END=50) I,J,K,Y
    L=1
    IF((J.EQ.B).OR.(J.EQ.M)) L=2
    X(K-26,I-71,L)=Y
    GO TO 1
50  DO 55 L=1,2
    DO 55 I=1,42
55  WRITE(12) (X(I,J,L),J=1,109)
    CALL EXIT
100 FORMAT(I4,I3,I7,A1,I11,I2,I17,F9.4)
END
/*

```

KNOWN ERRORS IN PROGRAM

1. Change irriration to irradiation in subroutines
OUTPUT, SPCASE, HEAT, SHIELD and ZAP.

APPENDIX 3

Description of Input

Input to ZAP is done on alphanumeric coded cards. At least one card must be chosen per group. Comments may be placed on any input card when it does not overlay an option field. The ",END" where specified may be extended into comments (i.e., Group 7 last card may be ...TIME 09,END COOLING TIMES). Comment cards may be placed anywhere in data deck. They are denoted by a dollar sign in column one and any legal character including blank other than a dollar sign in column two.

It will be noted that cards should be written exactly as shown because of spacing. ZAP checks key characters in input and not the entire card. Therefore misspelling will not necessarily cause errors. Key position will be explained below. CAUTION: begin all cards in column 1 as this position is always checked. "END" or ",END" must be properly placed to denote end of sections.

Lower case "b" denotes blank. Lower case "x" denotes data. "x" fields without decimal point are fixed point numbers and must be right justified. "x" fields with a decimal point are floating point fields and a decimal point may be anywhere within a field. "A" fields are alphanumeric character fields and letters must be right justified. "↓" denotes column 1; "]" denotes options; ")" denote necessary cards under an option (See INPUT CARDS).

1st card - has dollar signs in columns 1 and 2. User comments in columns 3-80.

Group 1 - Key checks are columns 12 and 33

If 1] is chosen the program calculates partial yields from the distribution shown in equation (9) of theory and chooses a mass yield curve as specified in column 53. Yield curves 1 and 3 denote fissioning of U-235 by 1 MeV and thermal neutrons respectively. Yield curves 2 and 4 denote fissioning of PU-239 by 2 MeV and thermal neutrons respectively.

If 2] is chosen, the program distribution is taken, but mass yields are read from cards. 2a) is a title card. It must be present and all 80 columns are a comment. 2b) denotes type of fission reaction constants used. If constants are desired from a yield library, (reference yields of preceding paragraph) choose desired curve data and place number in column 44 of card 2b1). If all data is to be read, use cards under 2b2). Key checks in 2b1) and 2b2) are columns 5-8. Key checks in 2b3) are columns 14, 16, 22, and 49. 2c) are input

yields per mass number. Card format is 5(E14.8,2X). There must be exactly 18 cards with an "END" in columns 78-80 of the last card.

If 3] is chosen, the program expects partial yields (i.e., yields per mass number and atomic numbers) as input. Card 3a) is a title card and is identical to 2a). There may be as many 3b cards as necessary to complete input. The ",END" must be present to denote end of section. Key positions in card type 3b) are columns 8, 20 and 30.

Group 2 - Key checks are columns 14, 15, 21, and 39.

The cards themselves are self explanatory. Progenitor yields are the different between the sum of the calculated partial yields in a chain and the actual input yield. These progenitors correspond to elements which decay instantaneously to the first member of the input chain and are thus treated.

The so-called "oblong table" is a table of calculated partial yields.

Group 3 - Key check is column 4

If 1] is chosen, the program skips calculation of mu and leaves mu's unchanged.

If 2] is chosen, the program uses λ for μ .

If 3] is chosen, the program calculates μ as shown from input information. Choose either Westcott averaged effective cross sections (card 3a1) or input cross sections (card 3a2). 3b gives neutron flux and options may be chosen. If any character other than a zero or blank is in columns 64, 66, or 68, the corresponding option of printing μ , cross sections, and/or λ s is effected. The entire text beginning with ",OPTIONS" need not be present. 3c) is a double purpose card used with both 3a1 and 3a2. See listing under "input cards" for details.

Group 4 - No positions checked

This card is a user inserted page title heading. Only 32 columns are used however all columns may be filled.

Group 5 - Key column checked is 9

These cards (from 1 through 11) are the cooling times (i.e., time after reactor shutdown). Columns 1-4 contain the time. Columns 5-8 contain the units of time.

Units are to be chosen from "SEC", "MIN", "DAYS", or "YRS", (right justified).

The cooling times must be sequentially numbered beginning with 01. These sequence numbers to be placed in appropriate columns. Again, ",END" denotes end of section.

Group 6 - Key columns checked are none

Identical to Group 5 except that these cards indicate irradiation time (i.e., time for which reactor is at power)

Group 7.- Key columns checked are 7, 11, and 15

This group indicates type of output desired. If no output is desired, choose card d]. If only selected isotopes are desired, choose card c] with as many b2 cards as necessary to obtain desired output. Card a1 determines how output is collected. "ALL ELEMENTS" outputs each isotope with non zero concentration by atomic number and sums concentrations for that atomic number. "MASS NUMBER ONLY" outputs sums of all concentrations by mass numbers. "ATOMIC NUMBER ONLY" outputs sums of all concentrations by atomic numbers. If selected isotopes are also

desired, use card b1 with as many b2 cards as necessary in addition to card a1 to obtain desired output.

Group 8 - Key columns checked are 11, 19, 38, 39, 41
44, 47, 50, 53, 56, 59, 62, 68 and 71

If 1] is chosen, no beta heating is calculated.

If 2] is chosen, beta heating energy from all elements for indicated cooling times is calculated and printed out.

If 3] is chosen, beta heating energy from isotopes listed on cards 3b) and for cooling times indicated is calculated and printed out. There may be as many 3b) type cards as necessary to obtain desired output.

Group 9 - Key columns checked are 10, 25, 35, 36, 38,
41, 44, 47, 48, 50, 53, 56, 59, 62, 65,
and 68.

If 1] is chosen, no gamma grouping is done.

If 2] is chosen, gamma grouping for all cooling times indicated is done. Card type 2b) gives group number, upper and lower energy limits in MeV. Group numbers must be sequential beginning with 01, but need not be listed in sequential order.

a) Title card (80 characters)

↓ As many as needed

b) AT.NO=xx,MASSbNO=xxx,YIELD=x,xxxxxxxxxE+xx,END

Note: ",END" denotes last card and must be present on last card
indicating end of section

Group 2

↓
1] YIELDboPTIONS=PRINTbPROGENITORS
↓
2] YIELDboPTIONS=PRINTbOBLONGbTABLE
↓
3] YIELDboPTIONS=PRINTbPROGENITORS,PRINTbOBLONGbTABLE
↓
4] YIELDboPTIONS=NONE

Group 3

↓
1] MU=OLD MU
↓
2] MU=LAMBDA
↓
3] MU=PHI*SIGMA+LAMBDA,DATA FOLLOWS-USE
↓
a) 1] WESTCOTT X-SECTIONS
↓
or
2] INDEPENDENT X-SECTIONS
↓
b) PHI=x.xxxxxxxxxxE+ xxN/SQ.CM/SEC,OPTIONS=PRINT(MU,X-SEC,LAMBDA)=
(x,x,x)

Note: ",OPTIONS=-----" is optional. If specified, place character other than a blank or zero in indicated position to get print-out of μ , cross section and/or lambdas. Blank or zero denotes option not effected.

c) The card type depends on type of cross sections chosen

<u>Westcott</u>	<u>Card Columns</u>	<u>Independent</u>	<u>Information</u>
Z=xx	1-4	Z=xx	Atomic number
A=xxxxA	7-12	A=xxxxA	Mass number and state
G= x.xxxxxxxxE+xx	14-31	X=x.xxxxxxxxE+xx	G factor or cross section
R= x.xxxxxxxxE+xx	34-51	no meaning	R factor
S= x.xxxxxxxxE+xx	54-71	no meaning	S factor
END	72-74	END	Denotes last card

Note: If S is not specified, it is assumed 1. If R is not specified, G is assumed to contain final correction factor. If A=000, the same correction factor or cross section is used for all mass numbers for specified atomic number.

Group 4

Page title card (columns 1-32)

example - 1bMEVbNEUTRONbFISSIONbOFbPU-239

Group 5

↓
x.xxAAAA=COOLINGbTIMEbxx,END

Note: One card for each cooling time up to eleven times. Times must be sequentially numbered in xx beginning with 01, ",END" must be on last cooling time card. "AAAA" denotes "MIN", "HRS", "DAYS", or "YRS".

Group 6

↓
x.xxAAAA=IRRADIATIONbTIMEbxx,END

Note: Same format as Group 5

Group 7

- a 1} OUTPUTbFORMAT=ALLbELEMENTS
↓
- 2] OUTPUTbFORMAT=MASSbNUMBERbONLY
↓
- 3] OUTPUTbFORMAT=ATOMICbNUMBERbONLY
↓
- b 1] OPTION:bELEMENTSbOFbSPECIALbINTEREST
↓
- 2] ATOMICbNO.=xx,MASSbNO.=xxx,END

Note: There may be as many b2 cards as desired. The ",END" must be

on the last card of this series.

If output is desired for all elements, choose one card from a]. depending on type of output desired. In addition, to output specific elements in a separate table, follow card a] with card b1 and as many b2 cards as required. If option is not desired, only an a] card should be input.

↓
c] NObOUTPUTbLIST-ELEMENTSbOFbSPECIALbINTERESTbONLY

If elements and isotopes are not desired and the user feels only certain isotopes are of interest, use card c] with as many b2 cards as is necessary to complete desired list.

↓
d] NObOUTPUT

If no output is desired, use card d]. It is then the users responsibility to obtain output from file 5.

Group 8

↓
1] NObBETAbHEATbGENERATION ALLbTIMES
↓
2] CALCULATEbBETAbHEATbFORbCOOLINGbTIMES=xx,xx,xx,...

Note: If "ALL TIMES" is chosen, program will calculate beta heating for all cooling times as specified in group 6. If specific

cooling times only are desired, list the sequence numbers of the cooling times desired in "xx,xx,xx,..." for as many times as desired. Last cooling time sequence number is followed by any legal character except a comma. The sequence numbers need not be in order.

3]

↓
a) SELECTIVEbBETA**b**HEAT**b**FORbCOOLINGbTIMES= ALLbTIMES
xx,xx,xx,...

Note: Same format as for 2) note.

↓
b) ATOMICbNO.=xx,MASS=xxx,END

Note: ",END" msut be on last card of 3b) input denoting end of deck. Use as many 3b cards as desired.

Group 9

↓
1] NObGAMMA**b**GENERATION

2] ↓
a) GAMMA**b**GENERATION**b**FORbCOOLINGbTIMES= ALLbTIMES
xx,xx,xx,...

Note: Same format as in Group 8, 2) note.

↓
b) GROUPbNO.=xx-UPPERbLIMIT=xxx.xxMEV,LOWERbLIMIT=xxx.xxMEV,END

Note: Group numbers must be from 1 to a maximum of 80 numbered

sequentially but not necessarily given in sequential order.

" ,END" must be on last group card denoting end of group input.

↓
Last Card \$\$ Comments (78 columns)

ZAP SAMPLE INPUT

SUPERZAP SAMPLE OUTPUT

FISSION PRODUCT SPECTRA FOR CA. 1 MEV NEUTRON FISSION OF U-235 ATOMS PER 200 ATOMS FISSIONED AFTER COOLING FOR 5.00 DAYS (IRRADIATION)

ELE	MASS	HALE-LIFE	5.00 SEC	10.0 SEC	15.0 SEC	30.0 SEC	1.00 MIN	15.0 MIN	3.00 HRS	1.00 DAYS	15.0 DAYS	120. DAYS	1.00 YRS	10.0 YRS	YRS
49 IN	121	750E-01	1.180E-03	1.171E-04	1.325E-06	1.325E-06	1.325E-06	1.325E-06	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	122	150E-01	1.533E-03	1.715E-04	1.303E-03	1.303E-03	1.303E-03	1.303E-03	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	123	150E-01	6.894E-03	7.732E-03	7.732E-03	7.732E-03	7.732E-03	7.732E-03	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	124	220E-02	3.511E-02	3.010E-03	3.870E-03	3.870E-03	3.870E-03	3.870E-03	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	125	150E-01	6.870E-02	6.441E-02	1.827E-02	1.827E-02	1.827E-02	1.827E-02	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	126	150E-01	4.870E-02	4.844E-02	1.272E-02	1.272E-02	1.272E-02	1.272E-02	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	127	300E-01	1.614E-02	1.187E-02	1.187E-02	1.187E-02	1.187E-02	1.187E-02	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	128	300E-01	1.130E-03	1.130E-03	1.130E-03	1.130E-03	1.130E-03	1.130E-03	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	129	150E-01	6.830E-03	6.830E-03	6.830E-03	6.830E-03	6.830E-03	6.830E-03	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	130	150E-01	4.832E-03	4.832E-03	4.832E-03	4.832E-03	4.832E-03	4.832E-03	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	131	150E-01	4.742E-04	2.017E-11	1.887E-20	1.887E-20	1.887E-20	1.887E-20	0.0	0.0	0.0	0.0	0.0	0.0	0.0
TOTALS FOR 49 IN =			2.717E-01	1.343E-01	8.608E-02	8.608E-02	8.608E-02	8.608E-02	4.039E-03	2.216E-03	1.228E-03	4.803E-03	5.663E-03	6.416E-03	0.0
50 SN	115	STARLE	3.099E-04	3.389E-09	3.389E-09	3.389E-09	3.389E-09	3.389E-09	8.389E-09	8.389E-09	8.389E-09	8.389E-09	8.389E-09	8.389E-09	8.389E-09
	117	STARLE	1.162E-05	1.162E-05	1.162E-05	1.162E-05	1.162E-05	1.162E-05	2.099E-04	2.099E-04	2.099E-04	2.099E-04	2.099E-04	2.099E-04	2.099E-04
	118	STARLE	1.133E-05	1.133E-05	1.133E-05	1.133E-05	1.133E-05	1.133E-05	6.401E-04	6.401E-04	6.401E-04	6.401E-04	6.401E-04	6.401E-04	6.401E-04
	119	STARLE	2.012E-07	1.162E-05	1.162E-05	1.162E-05	1.162E-05	1.162E-05	2.306E-02	2.306E-02	2.306E-02	2.306E-02	2.306E-02	2.306E-02	2.306E-02
	120	STARLE	2.356E-04	1.481E-04	1.481E-04	1.481E-04	1.481E-04	1.481E-04	2.821E-02	2.821E-02	2.821E-02	2.821E-02	2.821E-02	2.821E-02	2.821E-02
	121	STARLE	4.933E-04	1.481E-04	1.481E-04	1.481E-04	1.481E-04	1.481E-04	5.050E-02	5.050E-02	5.050E-02	5.050E-02	5.050E-02	5.050E-02	5.050E-02
	122	STARLE	1.679E-03	1.679E-03	1.679E-03	1.679E-03	1.679E-03	1.679E-03	5.485E-02	5.485E-02	5.485E-02	5.485E-02	5.485E-02	5.485E-02	5.485E-02
	123	STARLE	1.273E-03	1.273E-03	1.273E-03	1.273E-03	1.273E-03	1.273E-03	4.471E-12	4.471E-12	4.471E-12	4.471E-12	4.471E-12	4.471E-12	4.471E-12
	124	STARLE	1.144E-03	1.144E-03	1.144E-03	1.144E-03	1.144E-03	1.144E-03	1.067E-04	1.067E-04	1.067E-04	1.067E-04	1.067E-04	1.067E-04	1.067E-04
	125	STARLE	1.121E-02	1.121E-02	1.121E-02	1.121E-02	1.121E-02	1.121E-02	7.924E-02	7.924E-02	7.924E-02	7.924E-02	7.924E-02	7.924E-02	7.924E-02
	126	STARLE	1.121E-02	1.121E-02	1.121E-02	1.121E-02	1.121E-02	1.121E-02	6.366E-04	6.366E-04	6.366E-04	6.366E-04	6.366E-04	6.366E-04	6.366E-04
	127	STARLE	1.121E-02	1.121E-02	1.121E-02	1.121E-02	1.121E-02	1.121E-02	1.771E-01	1.771E-01	1.771E-01	1.771E-01	1.771E-01	1.771E-01	1.771E-01
	128	STARLE	1.121E-02	1.121E-02	1.121E-02	1.121E-02	1.121E-02	1.121E-02	4.964E-02	4.964E-02	4.964E-02	4.964E-02	4.964E-02	4.964E-02	4.964E-02
	129	STARLE	1.121E-02	1.121E-02	1.121E-02	1.121E-02	1.121E-02	1.121E-02	9.630E-09	9.630E-09	9.630E-09	9.630E-09	9.630E-09	9.630E-09	9.630E-09
	130	STARLE	1.121E-02	1.121E-02	1.121E-02	1.121E-02	1.121E-02	1.121E-02	5.149E-71	5.149E-71	5.149E-71	5.149E-71	5.149E-71	5.149E-71	5.149E-71
	131	STARLE	1.121E-02	1.121E-02	1.121E-02	1.121E-02	1.121E-02	1.121E-02	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	132	STARLE	1.121E-02	1.121E-02	1.121E-02	1.121E-02	1.121E-02	1.121E-02	1.233E-05	1.233E-05	1.233E-05	1.233E-05	1.233E-05	1.233E-05	1.233E-05
	133	STARLE	1.121E-02	1.121E-02	1.121E-02	1.121E-02	1.121E-02	1.121E-02	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	134	STARLE	1.121E-02	1.121E-02	1.121E-02	1.121E-02	1.121E-02	1.121E-02	0.0	0.0	0.0	0.0	0.0	0.0	0.0
TOTALS FOR 50 SN =			5.373E-02	5.373E-02	5.373E-02	5.373E-02	5.373E-02	5.373E-02	9.401E-01	5.576E-01	4.912E-01	4.691E-01	4.691E-01	4.691E-01	4.691E-01
51 SB	121	STARLE	2.254E-02	2.254E-02	2.254E-02	2.254E-02	2.254E-02	2.254E-02	2.254E-02	2.254E-02	2.254E-02	2.254E-02	2.254E-02	2.254E-02	2.254E-02
	122	STARLE	7.459E-03	7.459E-03	7.459E-03	7.459E-03	7.459E-03	7.459E-03	1.418E-09	1.418E-09	1.418E-09	1.418E-09	1.418E-09	1.418E-09	1.418E-09
	123	STARLE	1.162E-05	1.162E-05	1.162E-05	1.162E-05	1.162E-05	1.162E-05	6.401E-04	6.401E-04	6.401E-04	6.401E-04	6.401E-04	6.401E-04	6.401E-04
	124	STARLE	1.133E-05	1.133E-05	1.133E-05	1.133E-05	1.133E-05	1.133E-05	2.099E-04	2.099E-04	2.099E-04	2.099E-04	2.099E-04	2.099E-04	2.099E-04
	125	STARLE	2.012E-07	1.162E-05	1.162E-05	1.162E-05	1.162E-05	1.162E-05	2.306E-02	2.306E-02	2.306E-02	2.306E-02	2.306E-02	2.306E-02	2.306E-02
	126	STARLE	2.356E-04	1.481E-04	1.481E-04	1.481E-04	1.481E-04	1.481E-04	2.821E-02	2.821E-02	2.821E-02	2.821E-02	2.821E-02	2.821E-02	2.821E-02
	127	STARLE	4.933E-04	1.481E-04	1.481E-04	1.481E-04	1.481E-04	1.481E-04	5.050E-02	5.050E-02	5.050E-02	5.050E-02	5.050E-02	5.050E-02	5.050E-02
	128	STARLE	1.679E-03	1.679E-03	1.679E-03	1.679E-03	1.679E-03	1.679E-03	5.485E-02	5.485E-02	5.485E-02	5.485E-02	5.485E-02	5.485E-02	5.485E-02
	129	STARLE	1.273E-03	1.273E-03	1.273E-03	1.273E-03	1.273E-03	1.273E-03	4.471E-12	4.471E-12	4.471E-12	4.471E-12	4.471E-12	4.471E-12	4.471E-12
	130	STARLE	1.144E-03	1.144E-03	1.144E-03	1.144E-03	1.144E-03	1.144E-03	1.067E-04	1.067E-04	1.067E-04	1.067E-04	1.067E-04	1.067E-04	1.067E-04
	131	STARLE	1.121E-02	1.121E-02	1.121E-02	1.121E-02	1.121E-02	1.121E-02	7.924E-02	7.924E-02	7.924E-02	7.924E-02	7.924E-02	7.924E-02	7.924E-02
	132	STARLE	1.121E-02	1.121E-02	1.121E-02	1.121E-02	1.121E-02	1.121E-02	6.366E-04	6.366E-04	6.366E-04	6.366E-04	6.366E-04	6.366E-04	6.366E-04
	133	STARLE	1.121E-02	1.121E-02	1.121E-02	1.121E-02	1.121E-02	1.121E-02	1.771E-01	1.771E-01	1.771E-01	1.771E-01	1.771E-01	1.771E-01	1.771E-01
	134	STARLE	1.121E-02	1.121E-02	1.121E-02	1.121E-02	1.121E-02	1.121E-02	4.964E-02	4.964E-02	4.964E-02	4.964E-02	4.964E-02	4.964E-02	4.964E-02
TOTALS FOR 51 SB =			2.254E-02	2.254E-02	2.254E-02	2.254E-02	2.254E-02	2.254E-02	9.401E-01	5.576E-01	4.912E-01	4.691E-01	4.691E-01	4.691E-01	4.691E-01

GP. NO.	LOWER ENERGY BOUND (MEV)	UPPER ENERGY BOUND (MEV)	COOLING TIMES	SEC	SEC	SEC	MIN	MIN	DAYS	SEC IRRADIATION)	YRS	YRS
1	0.0	0.500	5.00E-03	1.00E-03	30.0	1.00E-03	1.00E-03	15.0	15.0	1.00E-09	0.0	10.0
2	0.501	1.000	1.273E-03	1.442E-03	1.442E-03	1.176E-03	5.417E-01	8.923E-05	8.004E-09	0.0	0.0	0.0
3	1.010	10.000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

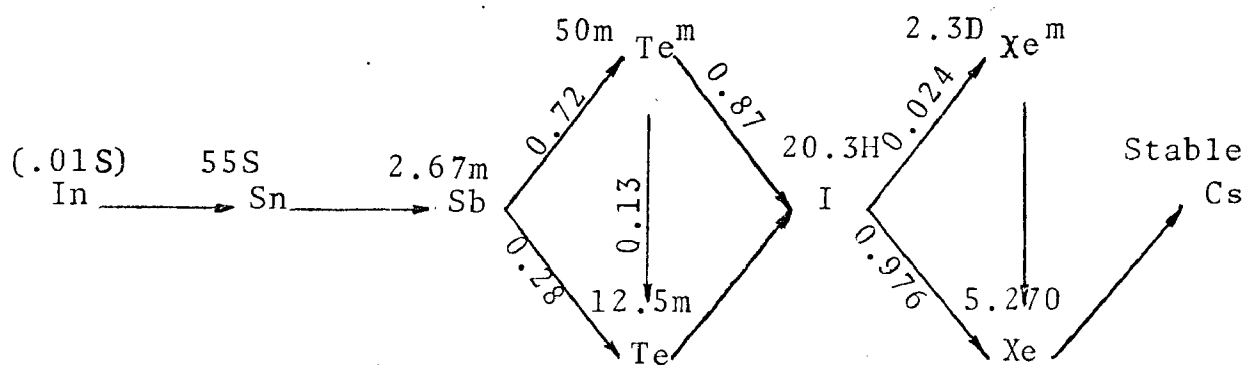
GAMMA ENERGY GROUP OPTION EFFECTED(5.00 SEC IRRADIATION)
OUTPUT IS ATOMS PER 200 ATOMS FISSIONED IN ENERGY GROUP

APPENDIX 4

Example of Program Procedure

The calculation of r_m and Y_m from basic nuclear data is demonstrated for the mass number 133 in a fission of U-235.

1. Reference 22 yields the decay chain;



2. The multibranch decay scheme may then be broken into the six equivalent straight line decay branches

$$1) \quad \text{In} \xrightarrow[r_1]{.015} \text{Sn} \xrightarrow[r_2]{.015} \text{Sb} \xrightarrow[r_3]{.015} \text{Te}^m \xrightarrow[r_4]{.020} \text{I} \xrightarrow[r_5]{.015} \text{Xe}^m \xrightarrow[r_6]{.626} \text{Xe} \xrightarrow[r_7]{.611} \text{Cs} \xrightarrow[r_8]{.015}$$

$$2) \quad \text{In} \xrightarrow[r_1]{.611} \text{Sn} \xrightarrow[r_2]{.611} \text{Sb} \xrightarrow[r_3]{.611} \text{Te}^m \xrightarrow[r_4]{.849} \text{I} \xrightarrow[r_5]{.611} \text{Xe} \xrightarrow[r_6]{.611} \text{Cs} \xrightarrow[r_7]{.611}$$

$$3) \quad \text{In} \xrightarrow{r_1} \text{Sn} \xrightarrow{r_2} \text{Sb} \xrightarrow{r_3} \text{Te}^m \xrightarrow{r_4} \text{Te} \xrightarrow{r_5}$$

$$\text{I} \xrightarrow{r_6} \text{Xe}^m \xrightarrow{r_7} \text{Xe} \xrightarrow{r_8} \text{Cs} \xrightarrow{r_9}$$

$$4) \quad \text{In} \xrightarrow{r_1} \text{Sn} \xrightarrow{r_2} \text{Sb} \xrightarrow{r_3} \text{Te}^m \xrightarrow{r_4}$$

$$\text{Te} \xrightarrow{r_5} \text{I} \xrightarrow{r_6} \text{Xe} \xrightarrow{r_7} \text{Cs} \xrightarrow{r_8}$$

$$5) \quad \text{In} \xrightarrow{r_1} \text{Sn} \xrightarrow{r_2} \text{Sb} \xrightarrow{r_3} \text{Te} \xrightarrow{r_4}$$

$$\text{I} \xrightarrow{r_5} \text{Xe}^m \xrightarrow{r_6} \text{Xe} \xrightarrow{r_7} \text{Cs} \xrightarrow{r_8}$$

$$6) \quad \text{In} \xrightarrow{r_1} \text{Sn} \xrightarrow{r_2} \text{Sb} \xrightarrow{r_3} \text{Tc} \xrightarrow{r_4}$$

$$\text{I} \xrightarrow{r_5} \text{Xe} \xrightarrow{r_6} \text{Cs} \xrightarrow{r_7}$$

Branching fractions, r_i , are then utilized in the nuclide concentrations for each decay branch. The terminating r_i of a branch is determined by the product of all branching fractions in that branch from the element of interest to the end of the branch. It is important to note that the r_i for a given nuclide must sum to unity when all the branches of the chain are considered. The equivalent branching fractions for the fourth branch of the chain are:

$$r_1 = r_8 = (1)(1)(.72)(.13)(1)(.976)(1) = .0913536$$

(Only 1st 3 digits noted on branches)

$r_1=r_2=r_3=r_6=r_7=r_8$ since all atoms beginning this chain must pass through the nuclides represented by 1,2,3,6,7, and 8 of this branch.

$$r_4 = r_3 / .72 = .126875$$

$$r_5 = r_3 / [(.72)(.13) + .28] = .2445128$$

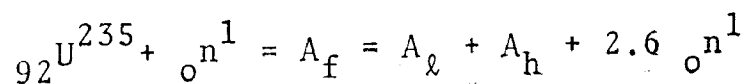
for a check

$$r_7 = r_6 / [6024(1) + .976] = .0913536$$

The equivalent branching fractions for branches 1 to 6, respectively, of the chain for I^{133} have the values: .0150336, .6113664, .0022464, .0913536, .00672, .27328, which sum to unity.

3. The total mass fission yield for $A=133$ from Table I is .06558.

4. To calculate the most probable atomic number for the heavy fission fragment, the nuclear reaction for the fission of U-235 by 1 MeV neutrons is used:



where: $A_f = 236$, $Z_f = 92$, $A_\ell + A_h = 236 - 2.6 = 233.4$

since $A_h = 133$, then $A_\ell = 100.4$

The most stable atomic number, for both heavy and light fission fragments is

$$Z_{sh} = 54.950 \quad (A_h = 133)$$

$$Z_{sl} = 43.916 \quad (A_l = 100.4) \text{ by interpolation}$$

The most probable atomic number for the heavy fission fragment is (from equation (7)).

$$Z_{ph} = 1/2[92 + (54.950 - 43.916)] = 51.515$$

5. The partial fission yields may then be calculated from each nuclide in the decay scheme from equation (9) in theory.

$$Y(Z,A) = Y(A)P(Z) = Y(A) \frac{1}{\sqrt{\pi}} e^{-[Z-Z_p(A)]^2}$$

Element	$Z-Z_p(A)$	$P(Z)$	$Y(Z,A)$
I_n	2.515	.0009	$.7319 \times 10^{-4}$
S_n	1.515	.0563	$.3726 \times 10^{-2}$
S_b	.515	.4327	$.2838 \times 10^{-1}$
T_e^m	.485	.4458/2	$.1462 \times 10^{-1}$
T_e	.485	.4458/2	$.1462 \times 10^{-1}$
I	1.485	.0622	$.4084 \times 10^{-2}$
X_e	2.485	.0011	$.3851 \times 10^{-4}$
C_s	3.485	.000015	$.1968 \times 10^{-6}$
			sum $\cong .06554$

The element T_e has two states in this branch. Therefore the yield is divided equally between them.

APPENDIX 5

File Construction

File 4 Fortran FT04F001 dsname=BEKLIB.RECA11.PERM4

This file is constructed as a direct access file for update purposes. There are two separate types of records (a and b) which are broken into 17 words and written in binary .

Record type a) (1 per chain

<u>Word No.</u>	<u>Information</u>
1	Mass or chain number
2	Maximum member of straight line decays in this chain
3-17	Maximum number of nodes in branches 1-15 respectively

Record type b) (1 per each node of entire chain)

<u>Word No.</u>	<u>Information</u>
1	Mass or chain number
2	Branch number
3	Node number
4	Lambda
5	Mu
6	Branching fraction
7	Partial yield
8	Element symbol
9	Element state
10	Atomic number
11	Number of elemental states of this element

<u>Word No.</u>	<u>Information</u>
12-17	Blank

Trailer Card (1 each)

1	999
---	-----

File 5 Fortran FT05F001 dsname=BEKLIB.RECB12.PERM5

There are 1050 of these records on a direct access file written in binary. There are 25 records allotted to each element with two states being contained in each 45 word record. Record 1050 is reserved for the element ^{124}Sb .

<u>Word No.</u>	<u>Information</u>
1	Atomic number
2	Element symbol
3	Mass number
4	State (1 of a possible 2)
5	Lambda (1 of a possible 2)
6-16	Final concentrations for cooling times 1-11 respectively (1 of a possible 2)
17	Check word - 0 if no data here, 1 if data is present
18	Check word - number of states in this record
19-31	Identical to words 4-16 for 2nd state if present
32	No. of possible decays (0,1,2) of 1st state

<u>Word No.</u>	<u>Information</u>
33	No. of possible decays (0,1,2) of 2nd state
34-35	Decay chain branching ratio to next element or elemental state (states 1 and 2 of this record respectively)
36-37	Beta decay energies to elements or element states from 1st state of this record
38-39	Gamma decay energy to elements or element states from 1st state of this record
40-43	Identical to words 36-39 for 2nd state of this record
44-45	Blank

File 9 Fortran FT09F001 dsname=BEKLIB.INSTREAM (in ZAP)

File 1 Fortran FT01F001 dsname=BEKLIB.INSTREAM (in-SUPERZAP)

This binary file contains 1 record of 112 words of constants and control parameters for the subprograms of SUPERZAP. They are:

<u>Word No.</u>	<u>Information</u>
1	Number of errors detected in ZAP
2	Code number specifying yield data to be used
3	Code number specifying progenitor option
4	Code number specifying oblong table option

<u>Word No.</u>	<u>Information</u>
5	Code number specifying how mu is to be calculated
6-13	Alphameric output page title given by user
14-35	Alphameric cooling times (2 words per time)
36-46	Numeric cooling times
47	Maximum number of cooling times (≤ 11)
48-58	Numeric burnup times
59-80	Alphanumeric burnup times (2 words per time)
81	Maximum number of burnup times (≤ 11)
82	Code number specifying output format
83	Code number specifying special output case
84	Maximum number of elements in special output case
85	Code number specifying type of beta heat calculations desired
86	Maximum number of cooling times to be calculated by beta heating
87-97	Cooling time code numbers to be used in beta heating calculations
98	Maximum number of beta heating cards read
99	Code specifying type of gamma calculations desired
100-110	Cooling time code numbers to be used by SHIELD

<u>Word No.</u>	<u>Information</u>
111	Maximum number of gamma energy groups

File 11 Fortran FT11F001 dsname=BEKLIB.ZAPDATA

This file contains all other data needed by SUPER ZAP that came from ZAP. It is written 80 character card images with number of records determined by options chosen.

File 12 Fortran FT12F001 dsname=BEKLIB.XSEC.PERM12

This machine language file contains the thermal cross sections for evaluating Westcott effective cross sections. This binary file contains 84 records of 109 words each.

Cross sections are arranged as follows:

```
DO X STATE = 1,2
DO X NATOM = 27,66
DO X MASS = 72, 161
```

X Cross section (NATOM, MASS, STATE)

File 14 Fortran FT14F001 dsname=BEKLIB.YLD1.PERM14

File 15 Fortran FT15F001 dsname=BEKLIB.YLD2.PERM15

File 16 Fortran FT16F001 dsname=BEKLIB.YLD3.PERM16

File 17 Fortran FT17F001 dsname=BEKLIB.YLD4.PERM17

These four files contain the information for each of the four resident mass yield curves. (See Appendix 5).

They each contain one record of 180 words. They are:

<u>Word</u>	<u>Information</u>
1	Compound fission nucleus mass number

<u>Word</u>	<u>Information</u>
2	Compound fission nucleus atomic number
3	Work 1 minus average neutrons released per fission
4-180	Yield in atoms normalized 200 atoms per mass numbers 4-180 respectively

File 18 Fortran FT180F001 dsname=BEKLIB.RECC18.PERM18

This binary file contains 3 files of 180 words each corresponding to the lowest, the highest, and the most stable atomic numbers per mass number 1 through 180 respectively.

File 3 Fortran FT03F001 SYSOUT=A

This file points to the output device to be used.

VITA

Bruce Earl Koopmann was born on May 6, 1942, in St. Louis, Missouri. He received his primary and secondary education in Affton, Missouri. He received his college education from University of Missouri - Rolla, in Rolla, Missouri where he received his Bachelor of Science Degree in Applied Mathematics in January, 1965.

He has been enrolled in Graduate School of the University of Missouri - Rolla since January, 1967 and has held a research assistantship from the University of Missouri - Rolla through the Nuclear Engineering Department since January, 1967.